

10/526022  
PCT/JP03/10891

日本国特許庁  
JAPAN PATENT OFFICE

29.09.03

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日  
Date of Application: 2002年 8月30日

出願番号  
Application Number: 特願2002-254010  
[ST. 10/C]: [JP2002-254010]

出願人  
Applicant(s): 関一

RECD 13 NOV 2003

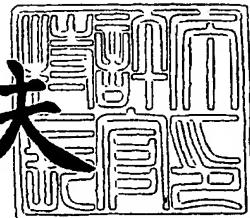
WIPO PCT

PRIORITY DOCUMENT  
SUBMITTED OR TRANSMITTED IN  
COMPLIANCE WITH  
RULE 17.1(a) OR (b)

2003年10月30日

特許庁長官  
Commissioner,  
Japan Patent Office

今井康夫



【書類名】 特許願

【整理番号】 RS002

【あて先】 特許庁長官殿

【国際特許分類】 G06F 15/00

【発明者】

【住所又は居所】 愛媛県松山市道後喜多町4番38号

【氏名】 関 一

【特許出願人】

【識別番号】 598003070

【住所又は居所】 愛媛県松山市道後喜多町4番38号

【氏名又は名称】 関 一

【手数料の表示】

【予納台帳番号】 057509

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 計算機システム

【特許請求の範囲】

【請求項 1】

各々のエントリにデータが書き込まれるようになっているデータ・ファイルと

各々のエントリに該データ・ファイルのエントリのアドレスが書き込まれるようになっている前進マッピング・ファイル及び完了マッピング・ファイルと、

実行待ちのオペレーションの内容を保持するバッファであるオペレーション・ウインドウと、

クロック・サイクル毎の該前進マッピング・ファイルの状態変更の内容が書き込めるようになっているFIFOキューの構成となっている状態変更キューと、

オペレーションを実行する機能ユニット群を具備し、

該前進マッピング・ファイル及び該完了マッピング・ファイルがスタックの内容を該スタックの先頭からエントリ・アドレス順に保持するようになっており、

該前進マッピング・ファイルの内容変更、その変更の内容の該状態変更キューへの書き込み、及び該前進マッピング・ファイルの該内容変更と組み合わさったオペレーション群の内容の該オペレーション・ウインドウへの書き込みを各々1クロック・サイクルで行うようになっており、

該オペレーション・ウインドウに保持されている未実行のオペレーションは各々out-of-orderで該機能ユニット群のいずれかによって実行されるようになっており、

該状態変更キューの先頭部の内容で示される該前進マッピング・ファイルの内容変更と組み合わさったオペレーション群に含まれるオペレーションが全て正常終了した後に、該状態変更キューの該先頭部の該内容に基づき該完了マッピング・ファイルの内容変更を行い、該状態変更キューから該先頭部を除外するようになっている計算機システム。

【発明の詳細な説明】

【0001】

**【産業上の利用分野】**

本発明は、高度の命令レベル並列性を実現することにより高性能化の可能な新規な構成の計算機方式に関するものである。

**【0002】****【従来の技術】**

従来、汎用の計算機として実用化してきた計算機方式は、スタック・マシンとレジスタ・ベースのマシンの2つに大きく分けることができる。スタック・マシンは、プログラムのコード・サイズが小さくてすみ、高速な割り込み処理やコンテキスト・スイッチングに優れているが、高速化が困難であった。他方、レジスタ・ベースのマシンは比較的高速化が容易であるが、その他の面ではスタック・マシンに劣る点が多い。

**【0003】****【発明が解決しようとする課題】**

スタック・マシンとレジスタ・ベースのマシンは、互いにプログラムの互換性が無く、また上述のように、いずれかが絶対的に優れているというものではない。

**【0004】**

本発明の目的は、スタック・マシンの命令形式ともレジスタ・ベースのマシンの命令形式とも親和性があり、かつ、高性能化の可能な計算機方式を実現することにある。

**【0005】****【課題を解決するための手段】**

本発明による計算機方式は、基本的にout-of-order実行の可能なスーパースカラ・アーキテクチャと同様なものであるが、ビット長の長い命令フォーマットを採用し、その中にオペレーション群の内容及び仮想のオペランド・スタックあるいは論理レジスタ群の状態変更の内容を記述するようになっている。

**【0006】**

本発明による計算機システムは、各々のエントリにデータが書き込まれるようになっているデータ・ファイルと、各々のエントリにデータ・ファイルのエントリのアドレスが書き込まれるようになっている前進マッピング・ファイル及び完了

マッピング・ファイルと、実行待ちのオペレーションの内容を保持するバッファであるオペレーション・ウインドウと、個々の命令によるマッピング・ファイルの状態変更の内容が書き込まれるようになっている状態変更キューと、各種オペレーションを実行する機能ユニット群と、割り付けられていないデータ・ファイルのエントリのアドレスを保持するフリー・リストとを具備する。

#### 【0007】

本発明の計算機システムは、1クロック・サイクル当たり1命令を発行して、オペレーション群の内容をオペレーション・ウインドウに書き込み、前進マッピング・ファイルの内容変更を行い、その変更の内容を状態変更キューに書き込むようになっている。結果データを生ずるオペレーションの場合、結果データを保持すべくデータ・ファイルのエントリが新たに割り付けられ、これがデスティネーション・レジスタとなる。また、各オペレーションにおいて、ソース・レジスタとなるのは、アドレスが前進マッピング・ファイルをアクセスして得られるデータ・ファイルのエントリ、あるいは、同じ命令に含まれるオペレーションにおいてデスティネーション・レジスタとして割り付けられるデータ・ファイルのエントリである。

#### 【0008】

オペレーション・ウインドウに保持されている未実行のオペレーションは各々out-of-orderでいずれかの機能ユニットによって実行されるようになっている。

#### 【0009】

状態変更キューの先頭部の内容と同じ命令の発行に基づくオペレーションが全て正常終了した後に、状態変更キューの先頭部の内容に基づき、その命令が発行された際の前進マッピング・ファイルの内容変更を再現すべく、完了マッピング・ファイルの内容変更を行い、状態変更キューから上記先頭部を除外するようになっている。

#### 【0010】

本発明による計算機システムは、スタック・モードとレジスタ・モードの2つのモードで動作することができる。スタック・ベースのプロセッサの機械語で記述されたプログラムを本発明の計算機システムの命令形式に変換して実行する場合

に、スタック・モードとなる。他方、RISCあるいはCISCのレジスタ・ベースのプロセッサの機械語で記述されたプログラムを本発明の計算機システムの命令形式に変換して実行する場合に、レジスタ・モードとなる。スタック・マシンあるいはレジスタ・マシンの複数の命令を、本発明の計算機システムにおける1つの命令に簡単に変換できるので、広範なプログラムを高速処理できる。

#### 【0011】

スタック・モードにおいては、マッピング・ファイルは、一種のスタックの内容を保持すべく用いられる。従来のスタックマシンにおいて、オペランド・スタックが ... , word3, word2, word1 (右端がスタックトップ) となっている状態は、本発明による計算機システムにおいて、マッピング・ファイルに保持される内容がエントリ・アドレス順に a, b, c, ... であれば、データ・ファイルにおいてエントリ・アドレスが a, b, c, ... である各エントリに、それぞれ word1, word2, word3, ... が保持されている状態に対応する。

#### 【0012】

レジスタ・モードにおいては、マッピング・ファイルは、レジスタ・ベースのスーパースカラ・プロセッサにおけるレジスタ・マッピング・テーブルと同等なものとして用いられる。

#### 【0013】

##### 【実施例】

以下に、本発明にかかる好ましい計算機システムについて説明する。

#### 【0014】

##### (1) 基本構成

まず、本実施例の計算機システムの基本的な構成について説明する。

#### 【0015】

図1は計算機システムのブロック図であって、10は命令キャッシュ、11はデータ・キャッシュ、12はデータ・バッファ、20は命令フェッチ・ユニット、21は命令発行ユニット、3aは前進マッピング・ファイル、3cは完了マッピング・ファイル、4は状態変更キュー、5はオペレーション・キュー、6はデータ・ファイル、7はフリー・リスト、80及び81は各々演算ユニット0及び1、8

2は分岐ユニット、83はロード／ストア・ユニットを表している。

### 【0016】

本発明の計算機システムは、実行待ちのオペレーションの内容を保持するバッファであるオペレーション・ウインドウを具備する必要がある。オペレーション・ウインドウを実現するには、基本的に集中型と分散型の2通りの方法があるが、本実施例の計算機システムは集中型のオペレーション・キューを備えるものとしている。

### 【0017】

本発明による計算機システムは、スタック・モードとレジスタ・モードの2つのモードで動作することができる。スタック・ベースのプロセッサの機械語で記述されたプログラムを本発明の計算機システムの命令形式に変換して実行する場合に、スタック・モードとなる。他方、RISCあるいはCISCのレジスタ・ベースのプロセッサの機械語で記述されたプログラムを本発明の計算機システムの命令形式に変換して実行する場合に、レジスタ・モードとなる。

### 【0018】

本発明の計算機システムの具備するマッピング・ファイルは、各々のエントリにデータ・ファイルのエントリのアドレスが書き込まれるような構成となっており、レジスタ・モードにおいては、レジスタ・ベースのout-of-order実行の可能なスーパースカラ・プロセッサにおけるレジスタ・マッピング・テーブルと同様に機能する。本発明において、マッピング・ファイルは、スタック・モードにおいても同等な機能を有するものである。

### 【0019】

#### (2) 命令形式

図2は本発明実施例の計算機システムの命令フォーマットを示す概念図であつて、スタック・モードにおいてもレジスタ・モードにおいても、個々の命令のフォーマットは、基本的に、フォーマット情報フィールド101、オペレーション・フィールド102及び状態変更フィールド103からなる。

### 【0020】

フォーマット情報フィールド (FIフィールド) では、当該命令の長さや、その命令が含むオペレーションの数などに関する情報が示される。

#### 【0021】

オペレーション・フィールド (Opフィールド) には、算術論理演算や分岐、コード／ストアなどからなるオペレーション群の内容が示される。本実施例においては、個々のオペレーションは、RISCプロセッサにおける命令と同様な形式で記述される。本実施例においては、1つの命令は最大4つまでのオペレーションを含むものとする（分岐オペレーションが含まれるような場合には、あえて1命令に4つのオペレーションを含める必要はない）。

#### 【0022】

状態変更フィールド (SMフィールド) では、マッピング・ファイルの内容の変更について示される。即ち、スタック・モードにおいては仮想のオペランド・スタック、レジスタ・モードにおいては仮想の論理レジスタ群の状態変更の内容が示される。後述するように、状態変更フィールド内部のフォーマットは、スタック・モードとレジスタ・モードで異なる。

#### 【0023】

結果データを生ずるオペレーションの場合、結果データを保持すべくデータ・ファイルのエントリが割り付けられるが、そのエントリの論理的な位置づけは、SMフィールドの内容によって示されることになる。

#### 【0024】

本実施例の計算機システムは、1サイクル当たり上記の形式の1命令を発行できるものである。

#### 【0025】

##### (2-1) スタック・モードにおける命令形式

以下に、スタック・モードにおける命令形式について説明する。

#### 【0026】

本実施例の計算機システムにおけるデータ・ファイルの各エントリが、仮想のスタック・マシンのオペランド・スタックの1エントリの保持するデータを保持するものとする。本実施例においては、マッピング・ファイルのアドレス0, 1, 2

, ... のエントリの内容をそれぞれ s0, s1, s2, ... とおくとき、エントリ・アドレスが s0, s1, s2, ... であるデータ・ファイルの各エントリにそれぞれ保持されるべきデータを順に並べることによって、仮想のスタック・マシンのオペランド・スタックの内容 (s0がスタックトップに対応) が構成されるべきものとする。

### 【0027】

例えば、 $A*X+B/X$ を計算し、その結果を変数Yとしてメモリにストアするプログラムは、スタック・マシンにおいては、次のように書ける。

```

LOAD <4>;ローカル変数4のデータをオペランド・スタックにプッシュする
DUP      ;スタックトップのデータをコピーし、そのコピーをスタックに積む
LOAD <1>;ローカル変数1のデータをオペランド・スタックにプッシュする
MUL      ;2変数分のデータをポップして乗算を行い、結果をプッシュする
SWAP      ;オペランド・スタックにおいて最上位とその次のデータを入れ替える
LOAD <2>;ローカル変数2のデータをオペランド・スタックにプッシュする
SWAP      ;オペランド・スタックにおいて最上位とその次のデータを入れ替える
DIV      ;2変数分のデータをポップして除算を行い、結果をプッシュする
ADD      ;2変数分のデータをポップして加算を行い、結果をプッシュする
STORE <5>;スタックトップのデータをローカル変数5にストアする

```

### 【0028】

ここで、変数A, B, X, Yの格納域を、それぞれローカル変数1, 2, 4, 5としている。

### 【0029】

上記プログラムがどのように本実施例の計算機システムにおけるスタック・モードの命令に変換されるかを説明する。

### 【0030】

上記プログラムを2つの命令群に分割して、個々の命令について、オペレーションの内容とマッピング・ファイルに加えられる変更の内容を分けて書くと、次のようなになる。

命令1a load f1, <4> ; +1: f1

命令1b add f2, f1, 0 ; +2: f1, f2  
命令1c load f3, <1> ; +3: f1, f2, f3  
命令1d mul f4, f2, f3 ; +2: f1, f4  
命令1e (swap) ; +2: f4, f1

---

命令2a load f1, <2> ; +1: f1  
命令2b (swap) ; +1: f1, s0  
命令2c div f2, f1, s0 ; 0: f2  
命令2d add f3, s1, f2 ; -1: f3  
命令2e store <5>, f3 ; -2:

### 【0031】

ここで、結果データを保持すべく新たに割り付けられるデータ・ファイルのエントリのアドレスを、各々の命令群において順にf1, f2, ... とおいている。本実施例においては、f1, f2, ... の各々は、各サイクルにおいてフリー・リストから取り出されるものに対応する。

### 【0032】

また、マッピング・ファイルに加えられる変更に関しては、スタックが何エントリ分成長するかという量と、スタックトップ近傍に加えられる変更内容（右端がスタックトップに対応）を示している。

### 【0033】

命令1aの意味は、ローカル変数<4>のデータをロードしf1に対応するデータ・ファイルのエントリに書き込み、その要素がマッピング・ファイルに保持されるスタックを1エントリ分成長させ、新たにスタックトップとなるエントリの内容をf1に対応するデータ・ファイルのエントリのアドレスとする、というものである。命令1bの意味は、f1に対応するデータ・ファイルのエントリのデータに0を加え、その結果データをf2に対応するデータ・ファイルのエントリに書き込み、その要素がマッピング・ファイルに保持されるスタックを1エントリ分成長させ、新たにスタックトップとなるエントリの内容をf2に対応するデータ・ファイルのエントリのアドレスとする、というものであるが、マッピング・ファイルに加え

られる変更については、命令1aによる変更内容も含めて記述している。その他の命令も同様である。

#### 【0034】

2つの命令群の各々について、その内容をまとめて、オペレーションとマッピング・ファイルの内容の変更に分けて記述すれば次のようになる。

```
Op{ load f1, <4>; add f2, f1, 0; load f3, <1>; mul f4, f2, f3 }
SM{ +2: f4, f1 }
```

---

```
Op{ load f1, <2>; div f2, f1, s0; add f3, s1, f2; store <5>, f3 }
SM{ -2: }
```

#### 【0035】

これが、上記プログラムを本発明実施例の計算機システムのスタック・モードの命令形式に変換した場合の、2命令の各々のOpフィールドとSMフィールドの内容である。このように、スタック・モードの命令形式では、SMフィールドにおいて、スタックの成長量と共に、更新後のスタックトップ近傍の内容を表す符号の系列が示されるようになっている。

#### 【0036】

##### (2-2) レジスタ・モードにおける命令形式

次に、レジスタ・モードにおける命令形式について説明する。

#### 【0037】

本実施例の計算機システムにおけるデータ・ファイルの各エントリが、仮想のレジスタ・マシンの1レジスタの保持するデータを保持できるものとする。本実施例においては、マッピング・ファイルのアドレス0, 1, 2, ... のエントリの内容をそれぞれr0, r1, r2, ... とおくとき、エントリ・アドレスがr0, r1, r2, ... であるデータ・ファイルの各エントリに、それぞれ仮想のレジスタ・マシンのレジスタR0, R1, R2, ... のデータが保持されるべきものとする。

#### 【0038】

例えば、メモリから変数A, B, Xのデータをレジスタ群にロードして、(A\*X+B/X)\*2を計算し、その結果を変数Yとしてメモリにストアするプログラムは、レジ

スタ・マシンにおいては、次のように書ける。

```

LOAD R1, 100[R0] ;(R0の中身+100)番地のデータをレジスタR1にロードする
LOAD R4, 400[R0] ;(R0の中身+400)番地のデータをレジスタR4にロードする
MUL R5, R1, R4 ;R5 = R1*R4
LOAD R2, 200[R0] ;(R0の中身+200)番地のデータをレジスタR2にロードする
DIV R4, R2, R4 ;R4 = R2/R4
ADD R5, R5, R4 ;R5 = R5+R4
MUL R5, R5, 2 ;R5 = R5*2
STORE 500[R0], R5 ;レジスタR5の内容を(R0の中身+500)番地にストアする

```

#### 【0039】

ここで、変数A, B, X, Yのメモリにおける格納域のアドレスを、R0の中身にそれぞれ100, 200, 400, 500を加えたものとしている。

#### 【0040】

上記プログラムがどのように本実施例の計算機システムにおけるレジスタ・モードの命令に変換されるかを説明する。

#### 【0041】

上記プログラムを4命令ずつから成るグループに分割して、個々の命令について、オペレーションの内容とマッピング・ファイルに加えられる変更の内容を分けて書くと、次のようになる。

```

命令1a load f1, 100[r0] ; 1:f1
命令1b load f2, 400[r0] ; 4:f2
命令1c mul f3, f1, f2 ; 5:f3
命令1d load f4, 200[r0] ; 2:f4

```

---

```

命令2a div f1, r2, r4 ; 4:f1
命令2b add f2, r5, f1 ; 5:f2
命令2c mul f3, f2, 2 ; 5:f3
命令2d store 500[r0], f3

```

#### 【0042】

ここで、スタック・モードの場合と同様に、結果データを保持すべく新たに割り付けられるデータ・ファイルのエントリのアドレスを、各々の命令群において順にf1, f2, ... とおいている。

#### 【0043】

命令1aの意味は、(r0に対応するデータ・ファイルのエントリのデータ+100)番地のデータをロードしf1に対応するデータ・ファイルのエントリに書き込み、マッピング・ファイルのアドレス1のエントリの内容をf1に対応するデータ・ファイルのエントリのアドレスに置き換える、というものである。命令1cの意味は、f1及びf2に各々対応するデータ・ファイルのエントリのデータを掛け合わせ、その結果データをf3に対応するデータ・ファイルのエントリに書き込み、マッピング・ファイルのアドレス5のエントリの内容をf3に対応するデータ・ファイルのエントリのアドレスに置き換える、というものである。その他の命令も同様である。

#### 【0044】

2つの命令群の各々について、4命令の内容をまとめて、オペレーションとマッピング・ファイルの内容の変更に分けて記述すれば次のようになる。

0p{load f1, 100[r0]; load f2, 400[r0]; mul f3, f1, f2; load f4, 200[r0]}  
SM{1:f1, 2:f4, 4:f2, 5:f3}

---

0p{div f1, r2, r4; add f2, r5, f1; mul f3, f2, 2; store 500[r0], f3}  
SM{4:f1, 5:f3}

#### 【0045】

これが、上記プログラムを本発明実施例の計算機システムのレジスタ・モードの命令形式に変換した場合の、2命令の各々の0pフィールドとSMフィールドの内容である。この2番目の命令に変換する際に、マッピング・ファイルのアドレス5のエントリの内容の置き換えが2つあるが、SMフィールドの内容としては最後のものだけが残されている。このように、レジスタ・モードの命令形式では、SMフィールドにおいて、内容を変更すべきマッピング・ファイルのエントリのアドレスと個々の変更内容を示す符号の組が列挙されるようになっている。

## 【0046】

## (3) スタック・モード

## (3-1) スタック・モードにおける動作に必要な機能と構成

ここでは、本実施例の計算機システムの各構成要素ごとに、スタック・モードの動作において利用される機能と構成について説明する。これらは、一部を除き、レジスタ・モードにおいても必要とされる機能と構成である。

## 【0047】

## (A) 命令フェッチ・ユニット

命令フェッチ・ユニットは、図示していないプログラムカウンタを具備しており、命令キャッシュから命令をフェッチし、命令発行ユニットに渡すようになっている。分岐の予測や分岐の実行も担う。

## 【0048】

## (B) 命令発行ユニット

命令発行ユニットは、命令フェッチ・ユニットから渡された命令の発行のために、その内容に基づき前進マッピング・ファイルやデータ・ファイルの操作、オペレーション・キーと状態変更キーへの書き込みを行うための各種信号を発生するようになっている。

## 【0049】

## (C) マッピング・ファイル

マッピング・ファイルは、各々のエントリにデータ・ファイルのエントリのアドレスが書き込まれるような構成となっている。

## 【0050】

本発明実施例の計算機システムのスタック・モードにおいては、マッピング・ファイル及びデータ・ファイルによって、スタック・マシンに具備されるべきスタックの最上位の部分が仮想的に構成されるようになっている。即ち、スタック・モードにおいては、マッピング・ファイルは、一種のスタックの内容を保持すべく用いられる。従来のスタックマシンにおいて、オペランド・スタックが ..., word3, word2, word1 (右端がスタックトップ) となっている状態は、本発明による計算機システムにおいて、マッピング・ファイルのアドレス0, 1, 2, ... のエン

トリの内容がそれぞれa, b, c, ... であれば、データ・ファイルにおいてエントリ・アドレスがa, b, c, ... である各エントリに、それぞれword1, word2, word3, ... が保持されている状態に対応する。オペランド・スタックの規模が大きくなると、残りの部分はデータ・バッファさらにはデータ・キャッシュに格納されるようになっている。

#### 【0051】

本発明の計算機システムは、前進マッピング・ファイル (AMF ; Advanced Mapping File) と完了マッピング・ファイル (CMF ; Completed Mapping File) の2つの同形のマッピング・ファイルを具備する。

#### 【0052】

本発明の計算機システムにおいては、命令が発行されるごとに、その命令の内容に応じてAMFの内容を変更する。すなわち、AMFは発行済みの全ての命令の内容を反映している。

#### 【0053】

他方、CMFは、プログラム上の順番で完了済みの全ての命令の内容を反映するものである。本発明の計算機システムは発行済みの命令に含まれるオペレーションのout-of-order実行を可能とするものであるが、CMFは、正確な例外処理を保証するため、in-orderで完了済みの全ての命令に基づく状態を構成するために存在するものである。

#### 【0054】

各マッピング・ファイルごとにスタックの要素を保持する最下位のエントリを示すボトム・ポインタと呼ぶレジスタが存在する。初期状態あるいはマッピング・ファイルが空の場合には、ボトム・ポインタの値は-1であるものとする。

#### 【0055】

図3は、本実施例の計算機システムにおける、2組のマッピング・ファイルとボトム・ポインタの関係を示す説明図である。2つのマッピング・ファイルAMF 3\_a及びCMF 3\_cは同数のエントリを有し、各マッピング・ファイルで各々のエントリに上から順に0, 1, 2, ... とアドレスが付けられている。斜線が施されていないAMF/CMFのエントリは、スタックの構成要素としてデータ・フ

ファイルのエントリのアドレスを保持しているものとする。図3に示すように、AMF及びCMFの各々に対して設けられたボトム・ポインタを、それぞれBa及びBcと名付けている。

#### 【0056】

(D) データ・ファイル (DF ; Data File)

データ・ファイル (DF) は、各々のエントリに1語分のデータが書き込まれるような構成となっている。

#### 【0057】

本実施例においては、DFのエントリにはp00, p01, p02, ... のようにアドレスが付けられているものとする。

#### 【0058】

図4は、本実施例の計算機システムにおける、DF6の各々のエントリ6(i)の構成を示す説明図である。ここで、iはエントリのアドレスである。DF6の各々のエントリ6(i)はデータ・フィールド61(i)、及び書き込みフラグ (WF ; Write Flag) フィールド62(i)から成っている。

#### 【0059】

実際のDFのハードウェア上の構成は、上述の各フィールド別に設けられたレジスタ・ファイルの集合体である。

#### 【0060】

DFの各々のエントリのデータ・フィールドは、1語分のデータが書き込まれるようになっている。

#### 【0061】

DFの各々のエントリにおいて、WFフィールドは、データ・フィールドにデータの書き込みが完了していれば1、完了していなければ0が書き込まれているようになっている。

#### 【0062】

(E) オペレーション・キュー (OQ ; Operation Queue)

オペレーション・キュー (OQ) は、未完了の発行済命令に含まれるオペレーションの内容を保持するバッファであり、循環型のFIFOキューの構成となつ

ている。

#### 【0063】

図5は、本実施例の計算機システムにおける、OQ5の各々のエントリ5(i)の構成を示す説明図である。ここで、iはエントリのアドレスである。OQ5の各々のエントリ5(i)はオペレーション・フィールド50(i)、デスティネーション・フィールド51(i)、オペランド・フィールド52(i)、第1ソース・フィールド53(i)、第1ソース書込フラグ(SWF1)フィールド54(i)、第2ソース・フィールド55(i)、第2ソース書込フラグ(SWF2)フィールド56(i)、リポート・フィールド57(i)、及びディスパッチ・フラグ・フィールド58(i)から成っている。

#### 【0064】

OQの各々のエントリのオペレーション・フィールドはオペレーション・コードが書き込まれるようになっている。

#### 【0065】

OQの各々のエントリのデスティネーション・フィールドは、結果データを生ずるオペレーションの場合に、その結果データを保持すべく割り付けられるDFのエントリのアドレスが書き込まれるようになっている。

#### 【0066】

OQの各々のエントリのオペランド・フィールドは、命令のOpフィールド中にオペランド値が示されるようなオペレーションの場合に、そのオペランド値が書き込まれるようになっている。

#### 【0067】

OQの各々のエントリの第1、第2ソース・フィールドの各々は、オペレーションのソース・データを保持すべく割り付けられているDFのエントリのアドレスが書き込まれるようになっている。

#### 【0068】

OQの各々のエントリにおいて、第1、第2の各SWFフィールドは各々第1、第2ソース・フィールドに対応して設けられている。SWF1/2フィールドは第1/2ソース・フィールドに示されるDFのエントリにデータの書き込みが

完了していれば1、完了していないければ0が書き込まれているようになっている。

。

#### 【0069】

OQの各々のエントリのリポート・フィールドは、そのオペレーションを含む命令に対応する状態変更キューのエントリのアドレスと識別番号 (A, B, C, D のうちのいずれか) が書き込まれるようになっている。

#### 【0070】

OQの各々のエントリにおいて、ディスパッチ・フラグ・フィールドは、オペレーションがまだディスパッチされていなければ0、ディスパッチされていれば1が書き込まれているようになっている。

#### 【0071】

(F) 状態変更キュー (SMQ ; State Modification Queue)

状態変更キュー (SMQ) は、未完了の発行済命令の各々によるマッピング・ファイルの状態変更の内容を保持するバッファであり、循環型のFIFOキューの構成となっている。本実施例においては、SMQの各々のエントリは1つの命令に対応するものとなっている。

#### 【0072】

図6は、本実施例の計算機システムにおける、SMQ4の各々のエントリ4(i)の構成を示す説明図である。ここで、iはエントリのアドレスである。SMQ4の各々のエントリ4(i)は状態変更 (SM) フィールド40(i)、オペレーション・ステータスA (OS\_A) フィールド41(i)、オペレーション・ステータスB (OS\_B) フィールド42(i)、オペレーション・ステータスC (OS\_C) フィールド43(i)、及びオペレーション・ステータスD (OS\_D) フィールド44(i)から成っている。

#### 【0073】

実際のSMQのハードウェア上の構成は、上述の各フィールド別に設けられたレジスタ・ファイルの集合体である。

#### 【0074】

SMQの各々のエントリのSMフィールドは、対応する命令のSMフィールド

に示されたマッピング・ファイルの変更内容が書き込まれるようになっている。

### 【0075】

SMQの各々のエントリにおいて、OS\_A、OS\_B、OS\_C、OS\_Dの各フィールドは、各々対応するオペレーションの実行状態に関する情報が書き込まれているようになっている。本実施例においては、簡単のため、対応するオペレーションが存在しないか正常終了していれば1、その他の場合には0が書き込まれているものとする。

### 【0076】

#### (G) データ・バッファ

データ・バッファは、各々のエントリに1語のデータが書き込まれるようになっている循環型のバッファである。

### 【0077】

#### (H) 機能ユニット

本実施例の計算機システムは、図1に示すように、演算ユニット0及び1、分岐ユニット及びロード／ストア・ユニット（LSU；Load Store Unit）の4つの機能ユニットを具備するものとしている。これらは、各々パイプライン処理が可能で、互いに独立に並行して動作するものである。

### 【0078】

スタック・モードにおいては、LSUはデータ・バッファ及びデータ・キャッシュにアクセスすることができるようになっている。

### 【0079】

本発明実施例の計算機システムのスタック・モードにおいては、マッピング・ファイル及びデータ・ファイル（DF）によって、スタック・マシンに具備されるべきスタックの最上位の部分が仮想的に構成されるが、その下の部分がデータ・バッファ、さらにその下の部分がデータ・キャッシュに格納されるようになっている。LSUはデータ・バッファに高速にアクセスできるようになっており、アクセスすべき変数データがデータ・バッファに保持されている割合が大きいほど、より効率的な計算が可能となる。また、データ・バッファに適当な語数のデータ

を溜めておくようすることによって、後述するDF-データ・バッファーデータ・キャッシュの間のSpill/Fillの動作を効率的に行うことができる。

#### 【0080】

LSUは、最初のローカル変数へのポインタを保持する図示してないレジスタ（varsレジスタ）を具備する。本実施例の計算機システムにおいては、ローカル変数の格納域はデータ・バッファあるいはデータ・キャッシュにあるが、varsレジスタには、データ・キャッシュにおける相当するアドレス値が書き込まれているようになっている。つまり、全てあるいは一部のローカル変数のデータが実際にはデータ・バッファに保持されていても、各々のローカル変数に、全てのローカル変数をデータ・キャッシュにSpillしたと仮定した場合のデータ・キャッシュにおけるアドレス値を対応させることができる。ロード／ストア・オペレーションの処理において、LSUはvarsレジスタの値を用いてアドレス計算を行い、対象となるローカル変数の格納域がデータ・バッファかデータ・キャッシュかを判定し、その格納域にアクセスする。

#### 【0081】

LSUは、プログラム中に示されるロード／ストア・オペレーションを実行すると共に、オーバーフロー／アンダーフローの回避のため、AMF／CMF及びDFによって構成される仮想の部分スタックの最下位にあたるデータをデータ・バッファとの間で自動的にSpill/Fillするようになっている。

#### 【0082】

1語分のデータをDFからデータ・バッファにSpillするには、(a) AMFとCMFで各々のボトム・ポインタ（Ba、Bc）で示されるエントリの内容が一致している、(b)その一致する内容と同じDFのエントリのアドレスがいずれかのソース・フィールドに書き込まれたエントリがOQに保持されるキュー内に存在しない、という2つの条件が満たされていなくてはならない（そうでない場合は満たされるまで待つ）。条件(b)に関しては、命令に含まれるオペレーションのソース・レジスタとして用いられるDFのエントリのアドレスがマッピング・ファイル中に保持された状態で残るケースに一定の制約を設けることによって、常に満たされるようにすることもできる。命令体系にそのような制約を設けない場合

には、計算機システムは、OQにおいて各ソース・フィールドで上記一致する内容であるDFのエントリのアドレスを照合し、上記条件(b)に関するチェックを行うような機能を備えた構成としなければならない。上記2つの条件が満たされる場合、AMF/CMFのボトム・ポインタBa/Bcで示されるエントリの内容で示されるDFのエントリに書き込まれている1語分のデータをデータ・バッファにSpillすることができる。その際、Ba及びBcの値から1を引き、上記DFのエントリのアドレスをFLに登録する。

#### 【0083】

逆に、データ・バッファからDFに1語分のデータをFillするには、データ・バッファからFillすべき1語分のデータを取り出し、それにフリーなDFの1エントリを割り付け、そのデータ・フィールドに書き込む。WCFフィールドは1とする。さらに、ボトム・ポインタBa及びBcの値に1を加え、上記割り付けられるDFのエントリのアドレスを、AMF及びCMFの各々1を加えたボトム・ポインタ値で示されるエントリに各々書き込む。

#### 【0084】

データ・バッファとデータ・キャッシュの間でも、データ・バッファの空きに応じて適宜Spill/Fillの動作が行われるようになっている。

#### 【0085】

以上では、命令の発行/完了に伴うAMF/CMFの操作がないものとして、DFとデータ・バッファの間のSpill/Fillの動作について説明したが、命令の発行/完了とSpill/Fillの動作を合成し、同時に行えるような構成は容易に実現できる。

#### 【0086】

また、本発明の計算機システムは、DFとデータ・バッファの間及びデータ・バッファとデータ・キャッシュの間で、一度に複数語のデータをSpill/Fillできるような構成とすることも可能である。

#### 【0087】

##### (I) フリー・リスト (FL ; Free List)

フリー・リスト (FL) は、フリーな、即ち、割り付けられていないDFのエ

ントリのアドレスを保持するもので、本実施例においては、図7に示すように、命令完了用登録バッファ（E B I C ; Entry Buffer for Instruction Completion）71a、スタック・スピル用登録バッファ（E B S S ; Entry Buffer for Stack Spill）71b、デスティネーション用割付キュー（A Q D ; Allocation Queue for Destinations）72a、及びスタック・フィル用割付キュー（A Q S F ; Allocation Queue for Stack Fill）72bを備える。これらは各々適当な数のDFのエントリのアドレスを保持でき、A Q DとA Q S Fは循環型のFIFOキューの構成となっている。

#### 【0088】

初期状態においては、DFの各々のエントリのアドレスはA Q DとA Q S Fのどちらかに保持されている。新たにDFのエントリを割り付ける必要がある場合にFLからフリーなDFのエントリのアドレスが取り出されるが、命令の発行に伴いオペレーションの結果データを保持すべく割り付けるためにはA Q Dから、また、その要素がマッピング・ファイルに保持されるスタックのFillのためにはA Q S Fから取り出されるようになっている。逆に、割り付けが解除されるDFの各々のエントリについてはそのアドレスがFLに登録されるようになっているが、命令の完了に伴い割り付けが解除されるものはE B I Cに、また、その要素がマッピング・ファイルに保持されるスタックのSpillに伴い割り付けが解除されるものはE B S Sに書き込まれるようになっている。E B I C/E B S Sに書き込まれた各々のDFのエントリのアドレスは、すぐに、A Q D及びA Q S Fの保持する内容量に応じて、そのいずれかに移されるようになっている。

#### 【0089】

##### (3-2) スタック・モードにおける動作

ついで、本発明実施例の計算機システムのスタック・モードにおける動作を説明する。

#### 【0090】

本実施例の計算機システムは、基本的に、命令を、①命令フェッチ、②命令発行、③オペレーション実行、④命令完了の4段階で処理する。以下に各段階ごとに動作内容を説明する。

## 【0091】

## ① 命令フェッチ

この段階では、命令フェッチ・ユニットが命令キャッシュから命令を取り出すと共に、次にフェッチする命令のアドレスを決定する。次に命令をフェッチするのは通常次アドレス値からであるが、フェッチした命令が無条件分岐オペレーションを含むか、条件分岐オペレーションを含み分岐すると予測した場合、分岐予測が外れた場合、あるいは例外が発生した場合には、フェッチするアドレス値を変更する。

## 【0092】

## ② 命令発行

この段階では、発行する命令のOpフィールドの内容に基づき命令に含まれるオペレーションの内容がオペレーション・キュ (OQ) に書き込まれ、SMフィールドの内容に基づき前進マッピング・ファイル (AMF) 及びそのボトム・ポインタBaの内容が変更され、その変更の内容が状態変更キュ (SMQ) に書き込まれる。この際、命令中のs0, s1, s2, ... の各々は、スタックの最上位、2番目、3番目、... に対応するが、それぞれ変更前のAMFのアドレス0, 1, 2, ... のエントリの内容に、f1, f2, ... の各々はそれぞれフリー・リスト (FL) から順に取り出されるDFのエントリのアドレスに、置き換えられる。

## 【0093】

AMF及びBaの内容変更に関しては、次のように行われる。まず、命令のSMフィールドに示される各符号が上記の要領でそれぞれ適切なDFのエントリのアドレスに置き換えられ、それぞれ対応するAMFのエントリに書き込まれる。ここで、AMFのアドレス0のエントリがスタックトップ即ち命令のSMフィールドにおいて右端に示される符号に対応し、以下アドレス順にAMFのエントリとスタックの要素が対応する。命令のSMフィールドで変更内容が明示されない部分については、スタックの成長量だけエントリの内容がシフトすることになる。即ち、スタックの成長量をgとおくと、AMFのアドレスiの内容が命令発行によりAMFのアドレス(i+g)に書き込まれることになる。また、ボトム・ポインタBaの値にはスタックの成長量が加えられる。

**【0094】**

発行される命令に含まれるオペレーションの内容が書き込まれるOQの各エントリにおいて、リポート・フィールドには、同じ命令に基づく書き込みが行われるSMQのエントリのアドレスと識別番号（順にA, B, C, Dと機械的に付すものとする）が書き込まれる。OQのエントリのディスパッチ・フラグ・フィールドには0が書き込まれる。

**【0095】**

命令の発行に伴い（f1, f2, f3, ...に対応する）新たに割り付けられるDFのエントリにおいては、WFフィールドに0が書き込まれる。

**【0096】**

ストア・オペレーションの場合には、その内容が、OQに書き込まれると共に、LSUに送られる。

**【0097】**

OQの書き込みが行われるソース・フィールドに対応するSWFフィールドはまず0とされるが、s0, s1, s2, ...のいずれかから置き換えられたDFのエントリ・アドレスのソース・フィールドへの書き込みの各々に関しては、次サイクルにおいて、そのアドレスのDFのエントリのWFフィールドの内容が読み出され、これが1であれば、OQの対応するSWFフィールドが1に変更されるようになっている。

**【0098】**

発行される命令に基づき書き込みが行われるSMQのエントリにおいて、対応するオペレーションの存在するオペレーション・ステータス・フィールドには未実行を意味する0が書き込まれ、その他には1が書き込まれる。

**【0099】****③ オペレーション実行**

OQに保持される未実行のオペレーションは、実行可能となったものから、適切な機能ユニットにディスパッチされ、処理されるようになっている。従って、オペレーションの実行順序はout-of-orderになる。

**【0100】**

OQにおいて、オペレーションに必要なソース・データが全てDFの該当するエントリに書き込み済であることが対応するSWFフィールドの内容によって確認されるエントリがあれば、その保持するオペレーションの実行のため、そのエントリの内容がDFをアクセスして得られるソース・データと共に適切な機能ユニットに送られるようになっている。この際、そのOQのエントリのディスペッチャ・フラグ・フィールドが1に変更される。

#### 【0101】

ロード・オペレーションや算術論理演算等の結果データを生じるオペレーションに関しては、結果データが正常に得られれば、デスティネーションであるDFのエントリのデータ・フィールドに結果データが書き込まれ、WFフィールドが1に変更される。また、OQにおいて各ソース・フィールドで上記デスティネーションであるDFのエントリのアドレスが照合され、一致するソース・フィールドに対応するSWFフィールドが1に変更される。

#### 【0102】

ストア・オペレーションを書き込みの内容とするOQのエントリに関しては、同じ内容が命令発行段階においてLSUに送られている。命令発行段階において確定しなかったアドレス計算に必要なソース・データは、本実施例においては、そのDFへの書き込みがOQにおいて確認された直後に、LSUに送られるようになっているものとする。

#### 【0103】

いずれのオペレーションも、その実行が正常に終了すれば、リポート・フィールドの内容に基づき、対応するSMQのエントリのオペレーション・ステータス・フィールドが1に変更される。

#### 【0104】

ストア・オペレーションに関しては、アドレス計算はオペレーション実行段階においてout-of-orderで実行されるが、正確な例外処理を保証するために、ストアの実行は命令完了段階において行われる。従って、ストア・オペレーションの場合、ストア・データとストア・アドレスが共に確定すれば、対応するSMQのエントリのオペレーション・ステータス・フィールドが1に変更されるようになって

いる。

### 【0105】

あるオペレーションの処理において例外事象が発生した場合には、その情報が、対応するSMQのエントリのオペレーション・ステータス・フィールドに書き込まれると共に、命令フェッチ・ユニットに通知される。

### 【0106】

#### ④ 命令完了

命令の完了はプログラム上の順番で行われる。

### 【0107】

SMQのキューの先頭のエントリにおいて、全てのオペレーション・ステータス・フィールドの内容が1である、あるいはそうなると、対応する命令の完了が可能となる。命令の完了を実行するには、SMQの先頭のエントリの内容に基づきCMFおよびBcの内容を変更し、SMQにおいて上記先頭のエントリをキューから除外する。ここで、命令の発行の際にAMFに対して行われた内容変更がCMFにおいて再現されている。

### 【0108】

また、命令の完了に伴い、割り付けを解除すべきDFのエントリのアドレスがそれぞれFLに登録される。スタック・モードにおいては、割り付けを解除すべきDFのエントリ群のアドレスは次の2つのグループからなる。命令の完了に伴う内容変更によりCMFにおいて保持されなくなるもの、及び、命令に含まれるオペレーションを保持していたOQのエントリ群のデスティネーション・フィールドに書き込まれていてCMFへの書き込みが行われないもの、である。

### 【0109】

ストア・オペレーションを含む命令を完了する場合には、LSUにストアの実行を依頼する。こうすれば、データがプログラム上の順番でストアされることが保証できる。

### 【0110】

以上が、本発明実施例の計算機システムのスタック・モードにおける動作の概要である。

## 【0111】

## (3-3) スタック・モードにおける動作例

ついで、具体的な動作例について説明する。いま、本実施例の計算機システムで、前述の $Y=A*X+B/X$ を計算する次の2命令からなるプログラム

命令1

```
0p { load f1, <4>; add f2, f1, 0; load f3, <1>; mul f4, f2, f3 }
SM { +2: f4, f1 }
```

命令2

```
0p { load f1, <2>; div f2, f1, s0; add f3, s1, f2; store <5>, f3 }
SM { -2: }
```

を実行するものとする。

## 【0112】

図8～図14は、本実施例の計算機システムにおいて、上記プログラムを処理する際の状態の変化を時系列に示した説明図であり、以下ではこの図をもとに詳細な動作を説明する。図8～図14において、DF6、OQ5及びSMQ4の各エントリの構成は、それぞれ図4、図5及び図6のものと同じである。マッピング・ファイルのボトム・ポインタが示すエントリより下のエントリや、FIFOキューの構成となっている構成要素のキューから外れるエントリなど、書き込み内容が意味あるものとして保持されていないような部分には、斜線が施されている。p##が記入されている箇所は、DFのいずれかのエントリのアドレスが書き込まれているが、本動作例の説明において留意する必要がないことを意味する。また、本動作例においては、説明を簡単にするため、DFとデータ・バッファの間のS pill/Fillの動作は行わないものとする。

## 【0113】

FLの構成要素であるデスティネーション用割付キュー (AQD) 72a及びAMF3aの当初の内容が図8に示されるようなものであるとしよう。ここで、循環型のFIFOキューの構成となっているAQD72aで、キューの先頭からの4エントリが次に発行される命令におけるf1, f2, f3, f4にそれぞれ対応することを示している。AMF3a及びCMF3cでは、各々のエントリに上から順

に0, 1, 2, ... とアドレスが付けられていて、それぞれs0, s1, s2, ... に対応するものとしている。

#### 【0114】

計算機システムが図8に示される状態にあるときに、命令1の発行が行われるものとする。

#### 【0115】

命令1の0pフィールドの内容に基づき命令に含まれるオペレーションの内容がOQに書き込まれ、SMフィールドの内容に基づきAMF及びBaの内容が変更され、その変更の内容がSMQに書き込まれる。この際、命令中のf1, f2, f3, f4の各々はそれぞれフリー・リストから順に取り出されるp26, p34, p42, p51に、置き換えられる。AMFの内容の変更に関しては、アドレス0, 1のエントリには、それぞれf1, f4から置き換えられるp26, p51が書き込まれ、それ以下の部分では、スタックの成長量(2)だけエントリの内容がシフトする(図8におけるAMFのアドレス0, 1, ... の各エントリの内容がアドレス2, 3, ... の各エントリにそれぞれ書き込まれる)。図9のOQで書き込み内容を示した4エントリにおいて、リポート・フィールドには、同じ命令1に基づく書き込みが行われるSMQのエントリのアドレス1と識別番号(順にA, B, C, D)が書き込まれている。また、これらのOQの4エントリのディスパッチ・フラグ・フィールドには0が書き込まれている。

#### 【0116】

f1, f2, f3, f4にそれぞれ対応して新たに割り付けられるDFのアドレスp26, p34, p42, p51の各エントリにおいては、WFフィールドを0とする。

#### 【0117】

SMQの書き込みが行われるエントリの各オペレーション・ステータス・フィールドには、いずれも対応するオペレーションが存在するので、0が書き込まれている。

#### 【0118】

こうして、計算機システムは図8の状態から図9に示されるような状態に至る。

。

**【0119】**

次のサイクルでは、命令1に続いて命令2の発行が行われる。

**【0120】**

命令1の場合と同様に、命令2の0pフィールドの内容に基づき命令に含まれるオペレーションの内容がOQに書き込まれ、SMフィールドの内容に基づきAMF及びBaの内容が変更され、その変更の内容がSMQに書き込まれる。この際、命令中のs0, s1は変更前（図9）のAMFのそれぞれアドレス0, 1のエントリの内容p26, p51に、f1, f2, f3の各々はそれぞれフリー・リストから順に取り出されるp16, p18, p57に、置き換えられる。SMフィールドに負のスタック成長量（-2）しか示されていないので、AMFに関しては、それだけエントリの内容がシフトする（図9におけるAMFのアドレス2, 3, … の各エントリの内容がアドレス0, 1, … の各エントリにそれぞれ書き込まれる）。

**【0121】**

f1, f2, f3にそれぞれ対応して新たに割り付けられるDFのアドレスp16, p18, p57の各エントリにおいては、WFフィールドを0とする。

**【0122】**

さらに、命令2はストア・オペレーションを含むので、その内容が、OQに書き込まれると共に、LSUに送られる。

**【0123】**

こうして、計算機システムは図10に示されるような状態に至る。

**【0124】**

OQに保持される未実行のオペレーションは、実行可能となったものから、適切な機能ユニットにディスパッチされ、処理される。

**【0125】**

図11は、数サイクルが経過して、命令1以前に発行された命令が全て完了し、命令1に含まれる4つのオペレーション全てが正常終了した時点の計算機システムの状態を示したものである。命令1に対応するSMQのアドレス1のエントリにおいて、全てのオペレーション・ステータス・フィールドが1となっている。また、この時点のCMF及びBcの内容は、命令1の発行直前（図8）のAMF

及びB aの内容と同じものとなっている（DFとデータ・バッファの間のSpill/Fillの動作が行われれば、命令発行時のB aの値と命令完了時のB cの値は一致しない）。この状態から命令1の完了が次のように実行される。

#### 【0126】

図11においてSMQのアドレス1のエントリがその時点のキューの先頭であるので、その内容に基づきCMF及びB cの内容が変更される。すなわち、B cの値に2が加えられ、CMFのアドレス0, 1のエントリにはそれぞれp26, p51が書き込まれ、それ以下の部分では、スタックの成長量(2)だけエントリの内容がシフトする。さらにSMQにおける上記先頭のエントリがキューから除外される。

#### 【0127】

また、命令1の完了に伴い割り付けを解除すべきDFのエントリのアドレスがFLに登録される。この場合、命令1に含まれるオペレーションを保持していたOQのエントリ群のデスティネーション・フィールドに書き込まれていてCMFへの書き込みが行われないp34, p42がFLの構成要素である命令完了用登録バッファ（EBIC）71aに書き込まれる。（この場合、命令の完了に伴う内容変更によりCMFにおいて保持されなくなるもの、はない。）こうして、計算機システムは図12に示されるような状態に至る。

#### 【0128】

図13は、さらに数サイクルが経過して、命令2に含まれる3つのオペレーションが正常終了し、最後のストア・オペレーションに関してはストア・データとストア・アドレスが共に確定済となった時点の計算機システムの状態を示したものである。EBICに書き込まれたp34, p42がAQDに移されている。

#### 【0129】

図13の時点から1サイクルで、命令2の完了が命令1の場合と同様に行われる。この場合、命令2がストア・オペレーションを含むので、そのストアの実行がLSUに依頼される。また、命令の完了に伴い割り付けを解除すべきDFのエントリのアドレスがFLに登録される。この場合、命令の完了に伴う内容変更によりCMFにおいて保持されなくなるものであるp51, p26と、命令2に含まれる

オペレーションを保持していたOQのエントリ群のデスティネーション・フィールドに書き込まれていてCMFへの書き込みが行われないものであるp16, p18, p57がEBCに書き込まれる。こうして、計算機システムは図14に示されるような状態に至る。

### 【0130】

以上で、本実施例の計算機システムにおいて $Y=A*X+B/X$ の計算が完了したことになる。

### 【0131】

#### (4) レジスタ・モード

##### (4-1) レジスタ・モードにおける動作に必要な機能と構成

(3-1) で述べた機能と構成は、一部を除き、レジスタ・モードにおいても必要とされるものである。ここでは、スタック・モードの場合との相違点について説明する。

### 【0132】

まず、マッピング・ファイルは、レジスタ・モードにおいては、レジスタ・ベースのスーパースカラ・プロセッサにおけるレジスタ・マッピング・テーブルと同等なものとして用いられる。

### 【0133】

従って、マッピング・ファイルのボトム・ポインタ、データ・バッファ、ロード／ストア・ユニット（LSU）のSpill/Fillに関する機能、フリー・リスト（FL）中のスタック・スピル用登録バッファ（EBS）及びスタック・ファイル用割付キュー（AQSF）などは、基本的に、必要ではない。

### 【0134】

#### (4-2) レジスタ・モードにおける動作

ついで、本発明実施例の計算機システムのレジスタ・モードにおける動作を説明する。

### 【0135】

本実施例の計算機システムは、レジスタ・モードにおいてもスタック・モードの場合と同様に、基本的に、命令を、①命令フェッチ、②命令発行、③オペレーシ

ヨン実行、④命令完了の4段階で処理する。以下に各段階ごとに動作内容を説明する。

#### 【0136】

##### ① 命令フェッチ

この段階では、命令フェッチ・ユニットが命令キャッシュから命令を取り出すと共に、次にフェッチする命令のアドレスを決定する。

#### 【0137】

##### ② 命令発行

この段階では、発行する命令のOpフィールドの内容に基づき命令に含まれるオペレーションの内容がオペレーション・キュー (O Q) に書き込まれ、SMフィールドの内容に基づき前進マッピング・ファイル (AMF) の内容が変更され、その変更の内容が状態変更キュー (SMQ) に書き込まれる。この際、命令中のr0, r1, r2, ... の各々はそれぞれ変更前のAMFのアドレス0, 1, 2, ... のエントリの内容に、f1, f2, ... の各々はそれぞれフリー・リスト (F L) から順に取り出されるDFのエントリのアドレスに、置き換えられる。

#### 【0138】

AMFの内容変更に関しては、次のように行われる。命令のSMフィールドに示されるAMFのエントリ・アドレスと符号の組の各々について、符号が上記の要領でそれぞれ適切なDFのエントリ・アドレスに置き換えられ、アドレスが示されるAMFのエントリに書き込まれる。

#### 【0139】

発行される命令に含まれるオペレーションの内容が書き込まれるO Qの各エントリにおいて、リポート・フィールドには、同じ命令に基づく書き込みが行われるSMQのエントリのアドレスと識別番号（順にA, B, C, Dと機械的に付すものとする）が書き込まれる。O Qのエントリのディスパッチ・フラグ・フィールドには0が書き込まれる。

#### 【0140】

命令の発行に伴い (f1, f2, f3, ... に対応する) 新たに割り付けられるDFのエントリにおいては、WFフィールドに0が書き込まれる。

**【0141】**

ストア・オペレーションの場合には、その内容が、OQに書き込まれると共に、LSUに送られる。

**【0142】**

OQの書き込みが行われるソース・フィールドに対応するSWFフィールドはまず0とされるが、r0, r1, r2, ... のいずれかから置き換えられたDFのエントリ・アドレスのソース・フィールドへの書き込みの各々に関しては、次サイクルにおいて、そのアドレスのDFのエントリのWFフィールドの内容が読み出され、これが1であれば、OQの対応するSWFフィールドが1に変更されるようになっている。

**【0143】**

発行される命令に基づき書き込みが行われるSMQのエントリにおいて、対応するオペレーションの存在するオペレーション・ステータス・フィールドには未実行を意味する0が書き込まれ、その他には1が書き込まれる。

**【0144】****③ オペレーション実行**

OQに保持される未実行のオペレーションは、スタック・モードの場合と同様に、実行可能となったものから、適切な機能ユニットにディスパッチされ、処理されるようになっている。

**【0145】****④ 命令完了**

命令の完了は、以下の点を除き、スタック・モードの場合と同様に、プログラム上の順番で行われる。

**【0146】**

まず、命令の発行の際にAMFに対して行われた内容変更がCMFにおいて再現されるが、それは、レジスタ・モードにおける命令のSMフィールド内部のフォーマット、即ち状態変更キュー（SMQ）の書き込みの形式に即して行われる。

**【0147】**

また、命令の完了に伴い、割り付けを解除すべきDFのエントリのアドレスが

それぞれFLに登録される。レジスタ・モードにおいては、割り付けを解除すべきDFのエントリ群のアドレスは次の2つのグループからなる。命令の完了に伴い内容が書き換えられるCMFの各エントリの変更前の内容を読み出したもの、及び、命令に含まれるオペレーションを保持していたOQのエントリ群のデスティネーション・フィールドに書き込まれていてCMFへの書き込みが行われないもの、である。

### 【0148】

以上が、本発明実施例の計算機システムのレジスタ・モードにおける動作の概要である。

### 【0149】

#### (4-3) レジスタ・モードにおける動作例

ついで、具体的な動作例について説明する。いま、本実施例の計算機システムで、前述の $Y=(A*X+B/X)*2$ を計算する次の2命令からなるプログラム

命令1

```
0p {load f1, 100[r0]; load f2, 400[r0]; mul f3, f1, f2; load f4, 200[r0]}
SM {1:f1, 2:f4, 4:f2, 5:f3}
```

命令2

```
0p {div f1, r2, r4; add f2, r5, f1; mul f3, f2, 2; store 500[r0], f3}
SM {4:f1, 5:f3}
```

を実行するものとする。

### 【0150】

図15～図21は、本実施例の計算機システムにおいて、上記プログラムを処理する際の状態の変化を時系列に示した説明図であり、以下ではこの図をもとに詳細な動作を説明する。図15～図21において、DF6、OQ5及びSMQ4の各エントリの構成は、それぞれ図4、図5及び図6のものと同じである。FI FIFOの構成となっている構成要素のキューから外れるエントリなどの書き込み内容が意味あるものとして保持されていないような部分には、斜線が施されている。p#が記入されている箇所は、DFのいずれかのエントリのアドレスが書

き込まれているが、本動作例の説明において留意する必要がないことを意味する。

#### 【0151】

FLの構成要素であるデスティネーション用割付キー (AQD) 72a及びAMF 3aの当初の内容が図15に示されるようなものであるとしよう。ここで、循環型の FIFOキーの構成となっているAQD 72aで、キーの先頭からの4エントリが次に発行される命令におけるf1, f2, f3, f4にそれぞれ対応することを示している。AMF 3a及びCMF 3cでは、各々のエントリに上から順に0, 1, 2, ... とアドレスが付けられていて、それぞれr0, r1, r2, ... に対応するものとしている。

#### 【0152】

計算機システムが図15に示される状態にあるときに、命令1の発行が行われるものとする。

#### 【0153】

命令1の0pフィールドの内容に基づき命令に含まれるオペレーションの内容がOQに書き込まれ、SMフィールドの内容に基づきAMFの内容が変更され、その変更の内容がSMQに書き込まれる。この際、命令中のr0は変更前(図15)のAMFのアドレス0のエントリの内容p12に、f1, f2, f3, f4の各々はそれぞれフリー・リストから順に取り出されるp26, p34, p42, p51に、置き換えられる。AMFの内容の変更に関しては、アドレス1, 2, 4, 5のエントリに、それぞれf1, f4, f2, f3から置き換えられるp26, p51, p34, p42が書き込まれる。図16のOQで書き込み内容を示した4エントリにおいて、リポート・フィールドには、同じ命令1に基づく書き込みが行われるSMQのエントリのアドレス1と識別番号(順にA, B, C, D)が書き込まれている。また、これらのOQの4エントリのディスパッチ・フラグ・フィールドには0が書き込まれている。

#### 【0154】

f1, f2, f3, f4にそれぞれ対応して新たに割り付けられるDFのアドレスp26, p34, p42, p51の各エントリにおいては、WFフィールドを0とする。(r0に対応するDFのアドレスp12のエントリでは、WFフィールドが1でデータ"1000"

がすでに書き込まれていたものとしている。)

#### 【0155】

SMQの書き込みが行われるエントリの各オペレーション・ステータス・フィールドには、いずれも対応するオペレーションが存在するので、0が書き込まれている。

#### 【0156】

こうして、計算機システムは図15の状態から図16に示されるような状態に至る。

#### 【0157】

次のサイクルでは、命令1に続いて命令2の発行が行われる。

#### 【0158】

命令1の場合と同様に、命令2のOpフィールドの内容に基づき命令に含まれるオペレーションの内容がOQに書き込まれ、SMフィールドの内容に基づきAMFの内容が変更され、その変更の内容がSMQに書き込まれる。

#### 【0159】

f1, f2, f3にそれぞれ対応して新たに割り付けられるDFのアドレスp16, p18, p57の各エントリにおいては、WFフィールドを0とする。

#### 【0160】

さらに、命令2は1つのストア・オペレーションを含むので、その内容が、OQに書き込まれると共に、LSUに送られる。

#### 【0161】

また、前サイクルの命令1の発行において、OQのソース・フィールドにr0から置き換えられたp12が書き込まれているので、そのアドレスのDFのエントリのWFフィールドの内容が読み出され、これが1であるので、OQにおいてp12が書き込まれているソース・フィールドに対応するSWFフィールドが1に変更される。

#### 【0162】

こうして、計算機システムは図17に示されるような状態に至る。

#### 【0163】

OQに保持される未実行のオペレーションは、実行可能となったものから、適切な機能ユニットにディスパッチされ、処理される。

#### 【0164】

図18は、数サイクルが経過して、命令1以前に発行された命令が全て完了し、命令1に含まれる4つのオペレーション全てが正常終了した時点の計算機システムの状態を示したものである。命令1に対応するSMQのアドレス1のエントリにおいて、全てのオペレーション・ステータス・フィールドが1となっている。また、この時点のCMFの内容は、命令1の発行直前（図15）のAMFの内容と同じものとなっている。この状態から命令1の完了が次のように実行される。

#### 【0165】

図18においてSMQのアドレス1のエントリがその時点のキューの先頭であるので、その内容に基づきCMFの内容が変更される。すなわち、CMFのアドレス1, 2, 4, 5のエントリにそれぞれp26, p51, p34, p42が書き込まれる。さらにSMQにおける上記先頭のエントリがキューから除外される。

#### 【0166】

また、命令1の完了に伴い割り付けを解除すべきDFのエントリのアドレスがFLに登録される。この場合、内容が書き換えられるCMFのアドレス1, 2, 4, 5の各エントリの変更前の内容（p02, p10, p24, p60）が読み出され、これらがFLの構成要素である命令完了用登録バッファ（EBIC）71aに書き込まれる。（この場合、命令1に含まれるオペレーションを保持していたOQのエントリ群のデスティネーション・フィールドに書き込まれていてCMFへの書き込みが行われないもの、はない。）こうして、計算機システムは図19に示されるような状態に至る。

#### 【0167】

図20は、さらに数サイクルが経過して、命令2に含まれる3つのオペレーションが正常終了し、最後のストア・オペレーションに関してはストア・データとストア・アドレスが共に確定済となった時点の計算機システムの状態を示したものである。EBICに書き込まれたp02, p10, p24, p60がAQDに移されている。

#### 【0168】

図20の時点から1サイクルで、命令2の完了が命令1の場合と同様に行われる。この場合、命令2がストア・オペレーションを含むので、そのストアの実行がLSUに依頼される。また、命令の完了に伴い割り付けを解除すべきDFのエントリのアドレスがFLに登録される。この場合、内容が書き換えられるCMFの各エントリの変更前の内容を読み出したものであるp34, p42と、命令2に含まれるオペレーションを保持していたOQのエントリ群のデスティネーション・フィールドに書き込まれていてCMFへの書き込みが行われないものであるp18がEBICに書き込まれる。こうして、計算機システムは図21に示されるような状態に至る。

#### 【0169】

以上で、本実施例の計算機システムにおいて $Y=(A*X+B/X)*2$ の計算が完了したことになる。

#### 【0170】

##### (5) 例外回復

ここでは、本発明の計算機システムにおける例外回復について説明する。

#### 【0171】

ある命令に含まれるオペレーションの実行において例外事象が発生した場合、その命令の発行時点の状態を回復する必要がある（但し、スタック・モードにおいては、すでに行われたデータ・ファイルDFとデータ・バッファの間のSpill/Fillの動作は取り消さない）。

#### 【0172】

そのためには、例外が発生した命令以降に発行された命令を全てキャンセルし、その時点の完了マッピング・ファイルCMF（及びそのボトム・ポインタBa）の内容を前進マッピング・ファイルAMF（及びそのボトム・ポインタBa）にコピーし、さらに、状態変更キューSMQのキューの先頭のエントリから上記例外が発生した命令に対応するエントリまでの各々の内容に基づきAMF（及びBa）の内容変更を行えばよい。

#### 【0173】

ある命令以降に発行された命令を全てキャンセルするには、オペレーション・

キューOQ及びSMQにおいてキャンセルされるべき命令群に対応する範囲のエントリを全てキューから除外し、キャンセルされるべきオペレーションのデスティネーション・レジスタとして割り付けられたDFのエントリのアドレスの各々をFLに戻せばよい。

#### 【0174】

キャンセルされるべきオペレーションのデスティネーション・レジスタとして割り付けられたDFのエントリのアドレスの各々をFLに戻すには基本的に2つの方法がある。即ち、まともに命令完了用登録バッファEBICに順次書き込む方法と、デスティネーション用割付キューAQDのキューの先頭を示すポインタを書き換えるというものである。後者の方針を採用する場合、AQDのキューの先頭を示すポインタ値として書き込まれる可能性がある値をどこかに保持しておく必要がある。それには、(a)SMQにそのためのフィールドを設け、命令の発行ごとに書き込む、(b)一種の履歴ファイルを設け、条件分岐オペレーションを含む命令の発行ごとに書き込む、というような方法が考えられる。

#### 【0175】

以上のように、本発明の計算機システムにおいては、例外が発生した命令が発行された時点のAMF(及びBa)の状態を、基本的に、回復することができる所以、正確な例外処理が可能である。

#### 【0176】

##### (6) その他の実施例

本発明の計算機システムは、上述の実施例に限られるものではなく、細部の構成の異なる様々な実施例が存在する。たとえば、以下のようなものをあげることができる。

#### 【0177】

##### (6-1) 実施例A

本発明の計算機システムは、従来のスタック・ベースあるいはレジスタ・ベースのプロセッサの命令形式に基づくプログラムを命令キャッシュに蓄えておき、実行時に命令発行段階の前段において、(2)で規定した命令形式に準じるよう変換するような構成とすることもできる。

## 【0178】

## (6-2) 実施例B

本発明の計算機システムは、個々の命令の発行の際に、その命令の完了時に割り付けの解除が必要になるDFのエントリのアドレス、あるいはそれを定めるための情報を、状態変更キューあるいは専用のキューにあらかじめ記入しておくような構成とすることもできる。

## 【0179】

## (6-3) 実施例C

共にボトム・ポインタを備えるスタック型の前進マッピング・ファイル及び完了マッピング・ファイルと、レジスタ型の前進マッピング・ファイル及び完了マッピング・ファイルを具備する構成とし、命令の状態変更フィールドで、スタック型のマッピング・ファイルの内容変更とレジスタ型のマッピング・ファイルの内容変更を共に示すことのできるような計算機システムも実現可能である。

## 【0180】

## (6-4) 実施例D

レジスタ・ベースのスーパースカラ・アーキテクチャにおいて考えられる様々な変形の多くが、本発明に基づく計算機システムにおいても適用できる。例えば、以下のようなものがあげられる。

- ① 各機能ユニットの入力段にそれぞれリザベーション・ステーションを設けることによってオペレーション・ウインドウを実現したもの。
- ② 整数データ用／浮動小数点データ用、あるいは汎用／マルチメディア用というようにデータ型別にデータ・ファイル、オペレーション・キュー、機能ユニット群、フリー・リストを設けたもの。
- ③ 複数組の前進マッピング・ファイル、完了マッピング・ファイル、オペレーション・キュー、状態変更キュー等を設け、複数のスレッドを並行して処理できる構成としたもの。

## 【0181】

## 【発明の効果】

本発明によれば、高性能の計算機システムを実現することができる。また、従

來のスタック・ベースあるいはレジスタ・ベースのプロセッサの機械語で記述されたプログラムは、本発明の計算機システムの命令形式に容易に変換できる。

【図面の簡単な説明】

【図 1】

本発明の一実施例の計算機システムの基本構成を示すブロック図である。

【図 2】

本発明の一実施例の計算機システムの命令フォーマットを示す概念図である。

【図 3】

本発明の一実施例のスタック・モードにおける、前進マッピング・ファイル、完了マッピング・ファイル、及び2つのボトム・ポインタの関係を示した説明図である。

【図 4】

本発明の一実施例におけるデータ・ファイル (D F) 6 の各々のエントリ 6 (i) の詳細な構成を示した説明図である。

【図 5】

本発明の一実施例におけるオペレーション・キュー (O Q) 5 の各々のエントリ 5 (i) の詳細な構成を示した説明図である。

【図 6】

本発明の一実施例における状態変更キュー (SMQ) 4 の各々のエントリ 4 (i) の詳細な構成を示した説明図である。

【図 7】

本発明の一実施例におけるフリー・リスト 7 の内部構成を示した説明図である。

【図 8】

図 8～図 14 は、本発明の一実施例のスタック・モードにおける一動作例の、サイクル毎の、前進マッピング・ファイル (AMF) 3 a、完了マッピング・ファイル (CMF) 3 c、状態変更キュー (SMQ) 4、オペレーション・キュー (O Q) 5、データ・ファイル (D F) 6、及びフリー・リスト (F L) 7 の内容を

具体的に示した説明図である。図8は当初の内容を示した図である。

【図9】

図8に続く段階の内容を示した図である。

【図10】

図8～図9に続く段階の内容を示した図である。

【図11】

図8～図10に続く段階の内容を示した図である。

【図12】

図8～図11に続く段階の内容を示した図である。

【図13】

図8～図12に続く段階の内容を示した図である。

【図14】

図8～図13に続く段階の内容を示した図である。

【図15】

図15～図21は、本発明の一実施例のレジスタ・モードにおける一動作例の  
、サイクル毎の、前進マッピング・ファイル (AMF) 3a、完了マッピング・フ  
ァイル (CMF) 3c、状態変更キー (SMQ) 4、オペレーション・キー  
(OQ) 5、データ・ファイル (DF) 6、及びフリー・リスト (FL) 7の内容  
を具体的に示した説明図である。図15は当初の内容を示した図である。

【図16】

図15に続く段階の内容を示した図である。

【図17】

図15～図16に続く段階の内容を示した図である。

【図18】

図15～図17に続く段階の内容を示した図である。

【図19】

図15～図18に続く段階の内容を示した図である。

【図20】

図15～図19に続く段階の内容を示した図である。

## 【図21】

図15～図20に続く段階の内容を示した図である。

## 【符号の説明】

- 3 a 前進マッピング・ファイル (AMF)
- 3 c 完了マッピング・ファイル (CMF)
- 4 状態変更キュー (SMQ)
- 5 オペレーション・キュー (OQ)
- 6 データ・ファイル (DF)
- 7 フリー・リスト (FL)
- 10 命令キャッシュ
- 11 データ・キャッシュ
- 12 データ・バッファ
- 20 命令フェッチ・ユニット
- 21 命令発行ユニット
- 30 a ボトム・ポインタ Ba
- 30 c ボトム・ポインタ Bc
- 71 a 命令完了用登録バッファ (EBIC)
- 71 b スタック・スピル用登録バッファ (EBSS)
- 72 a デスティネーション用割付キュー (AQD)
- 72 b スタック・ファイル用割付キュー (AQSF)
- 80 演算ユニット0
- 81 演算ユニット1
- 82 分岐ユニット
- 83 ロード／ストア・ユニット (LSU)

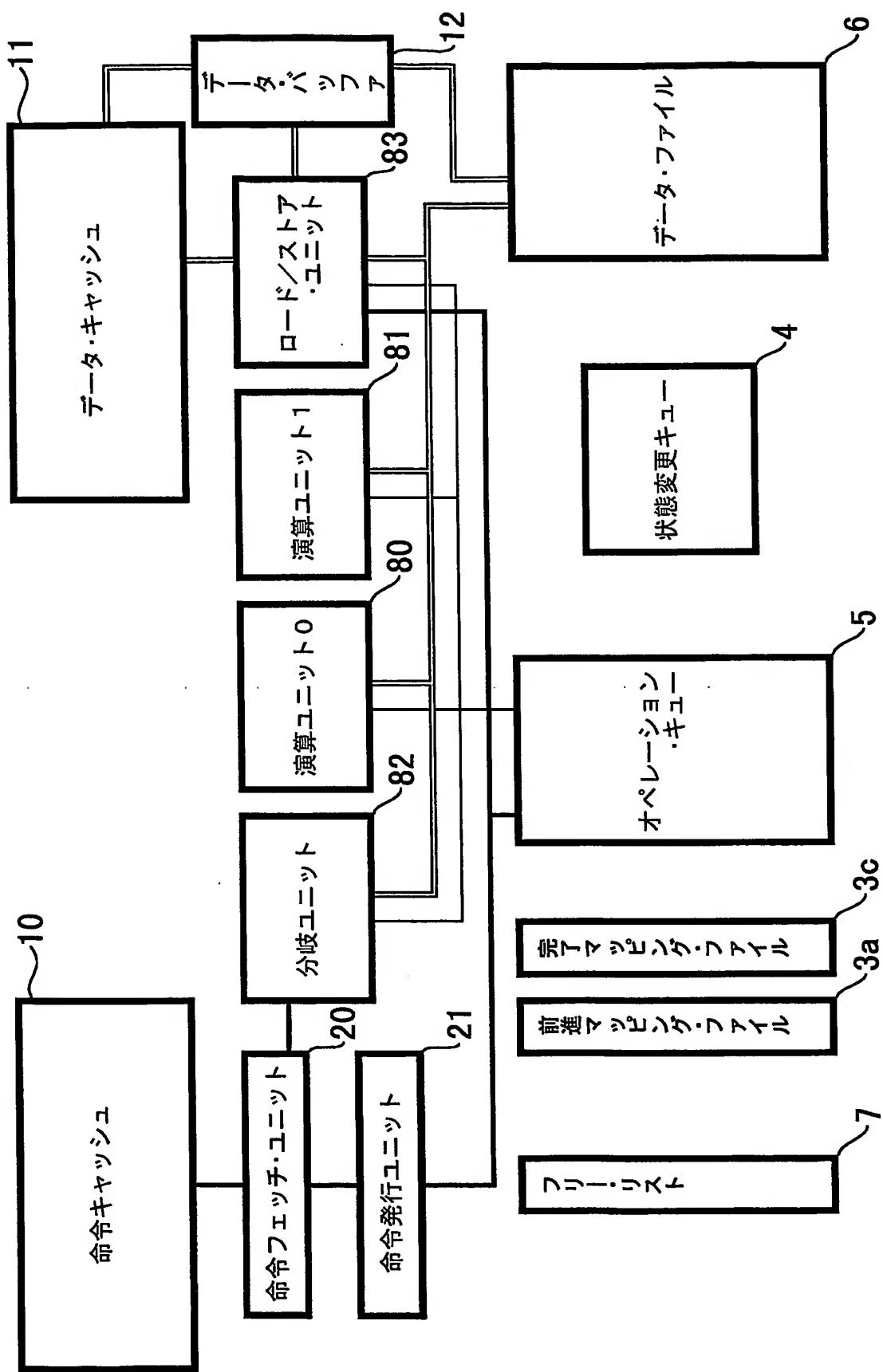
特願 2002-254010

ページ： 1/

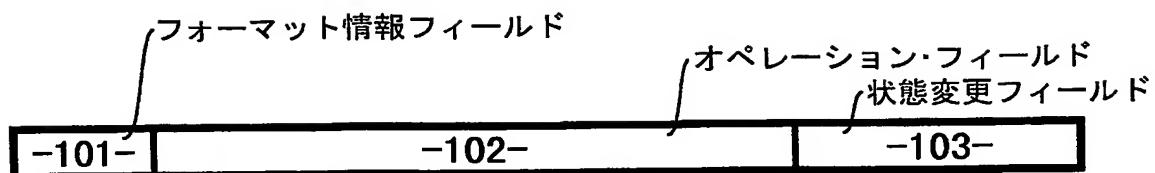
【書類名】 図面

【図 1】

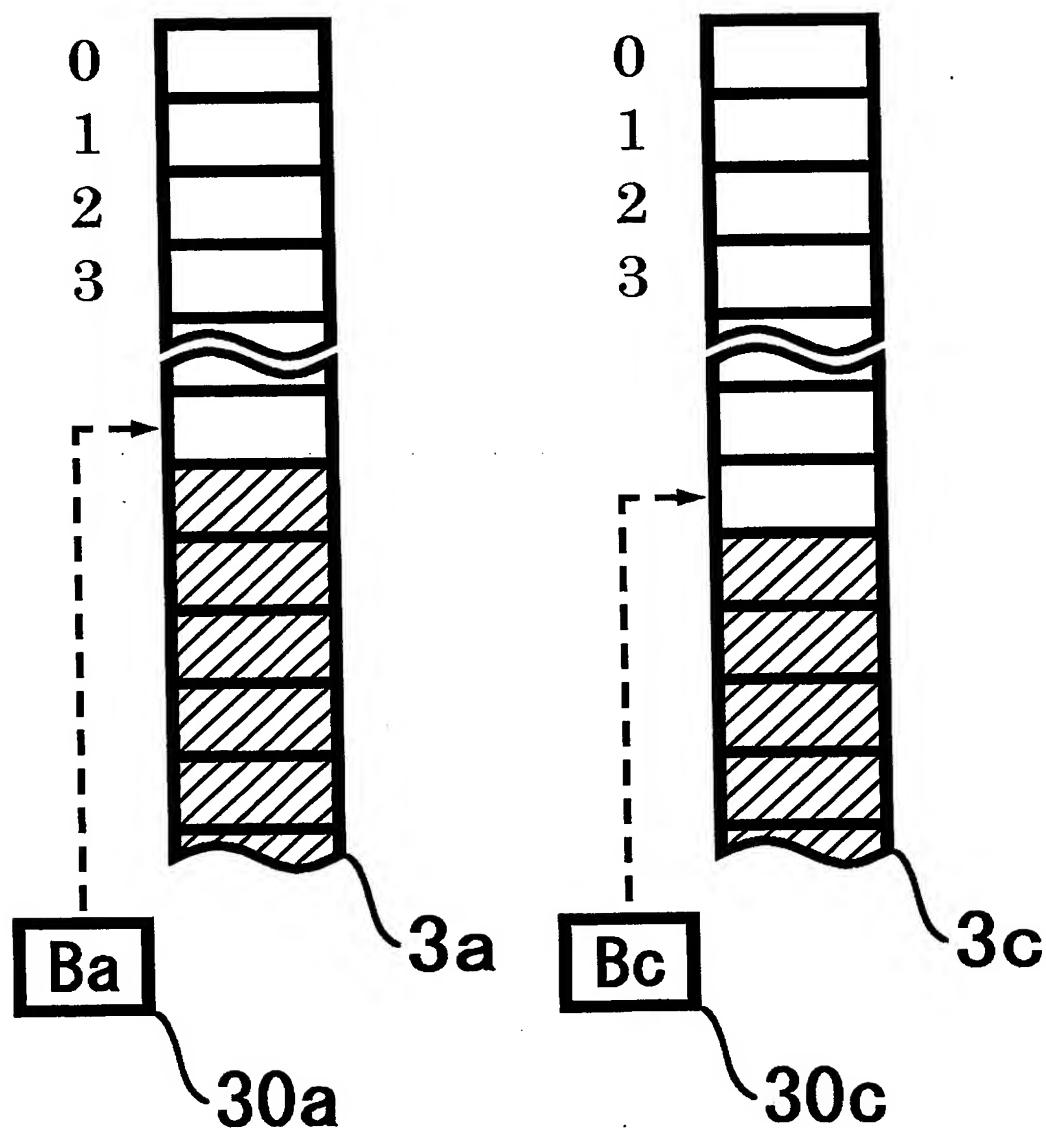
出証特 2003-3089730



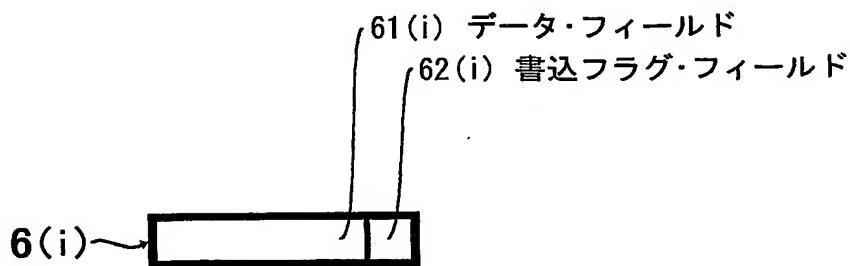
【図2】



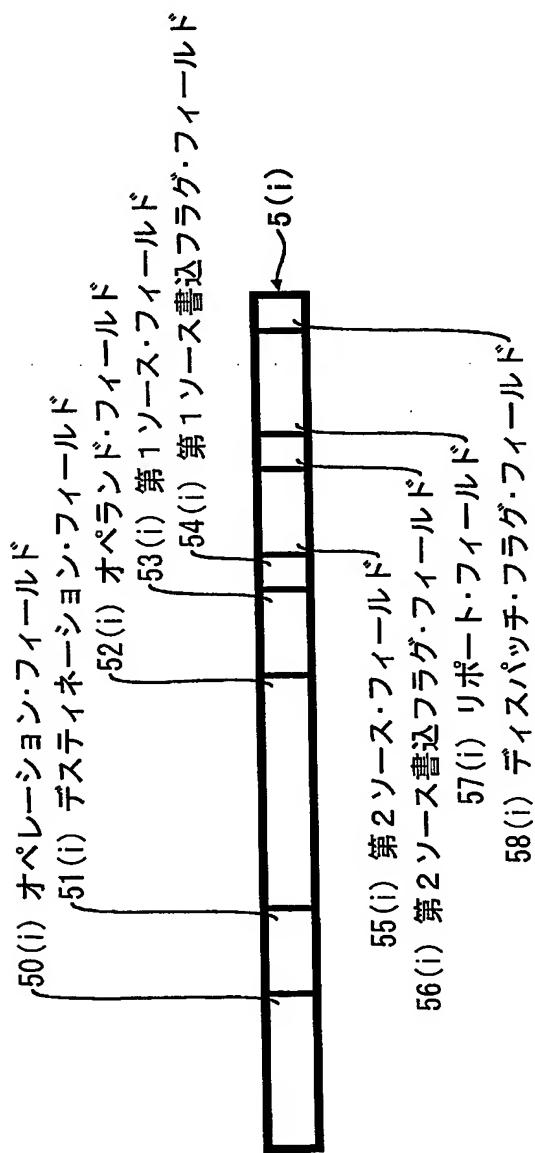
【図3】



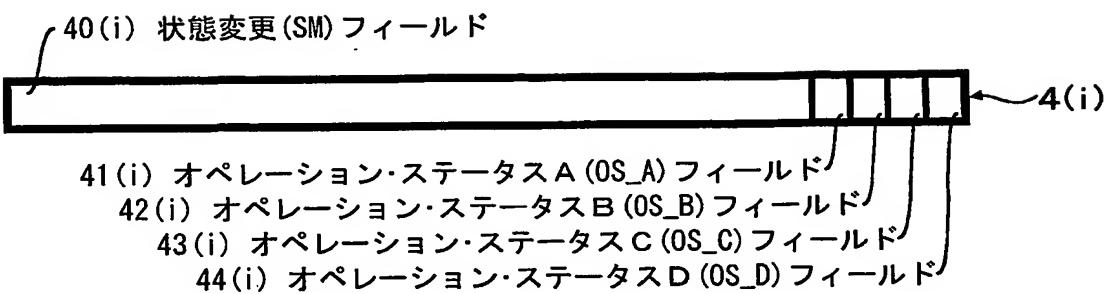
【図4】



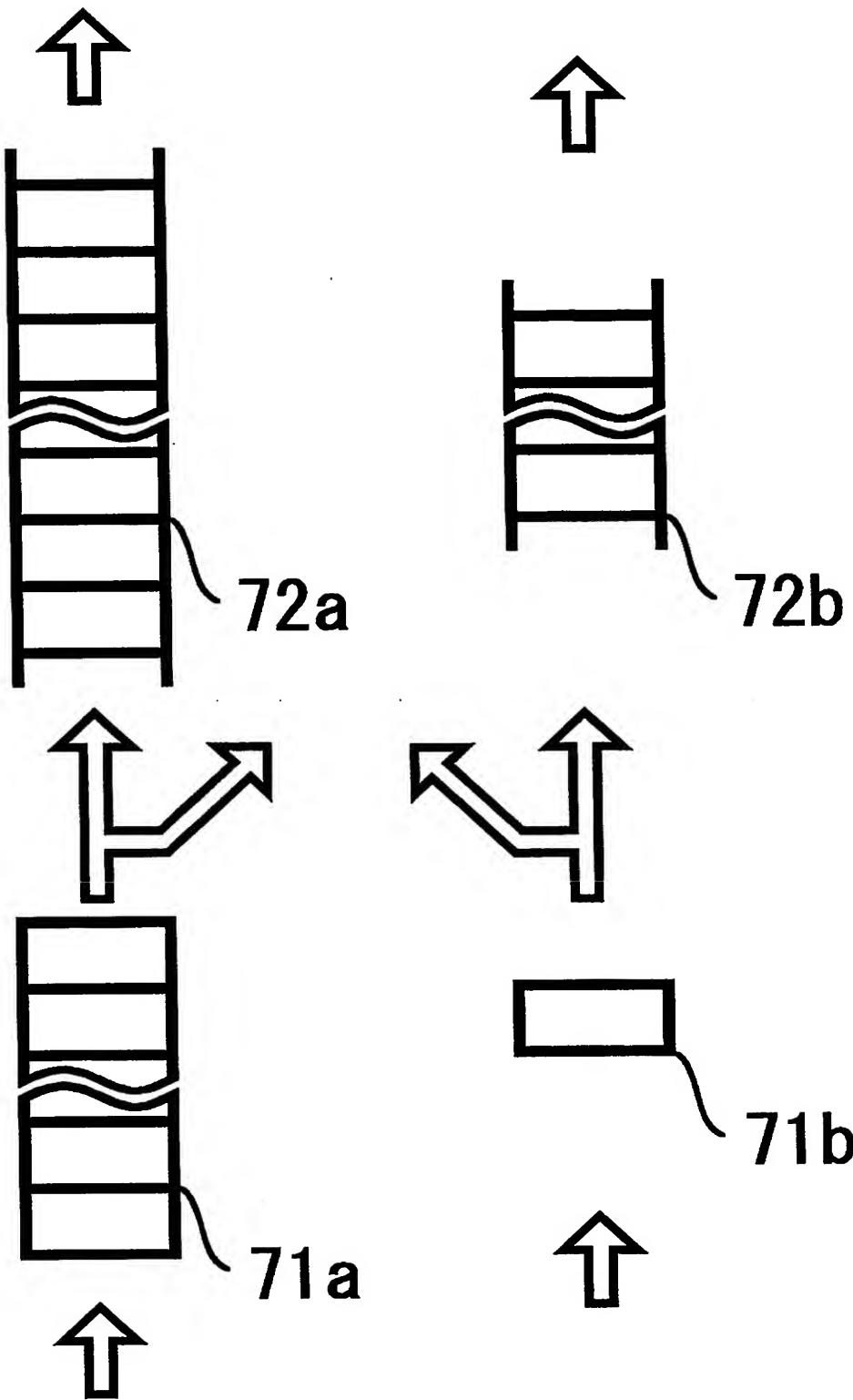
【図5】



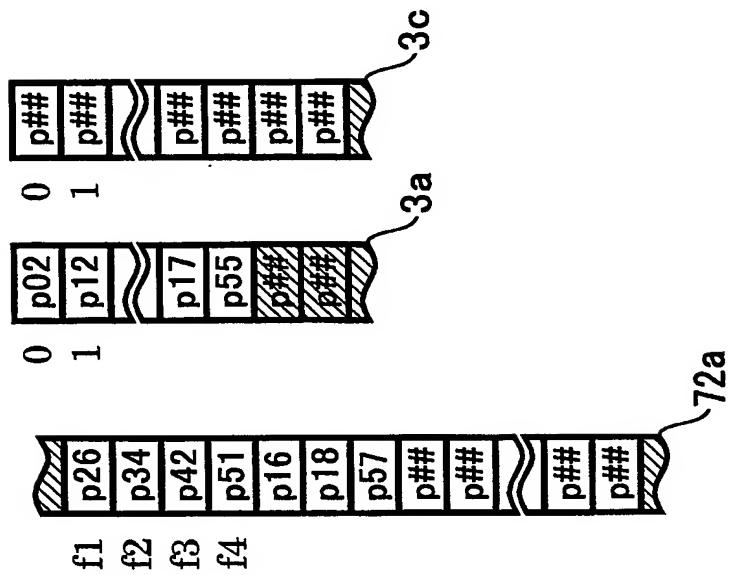
【図6】



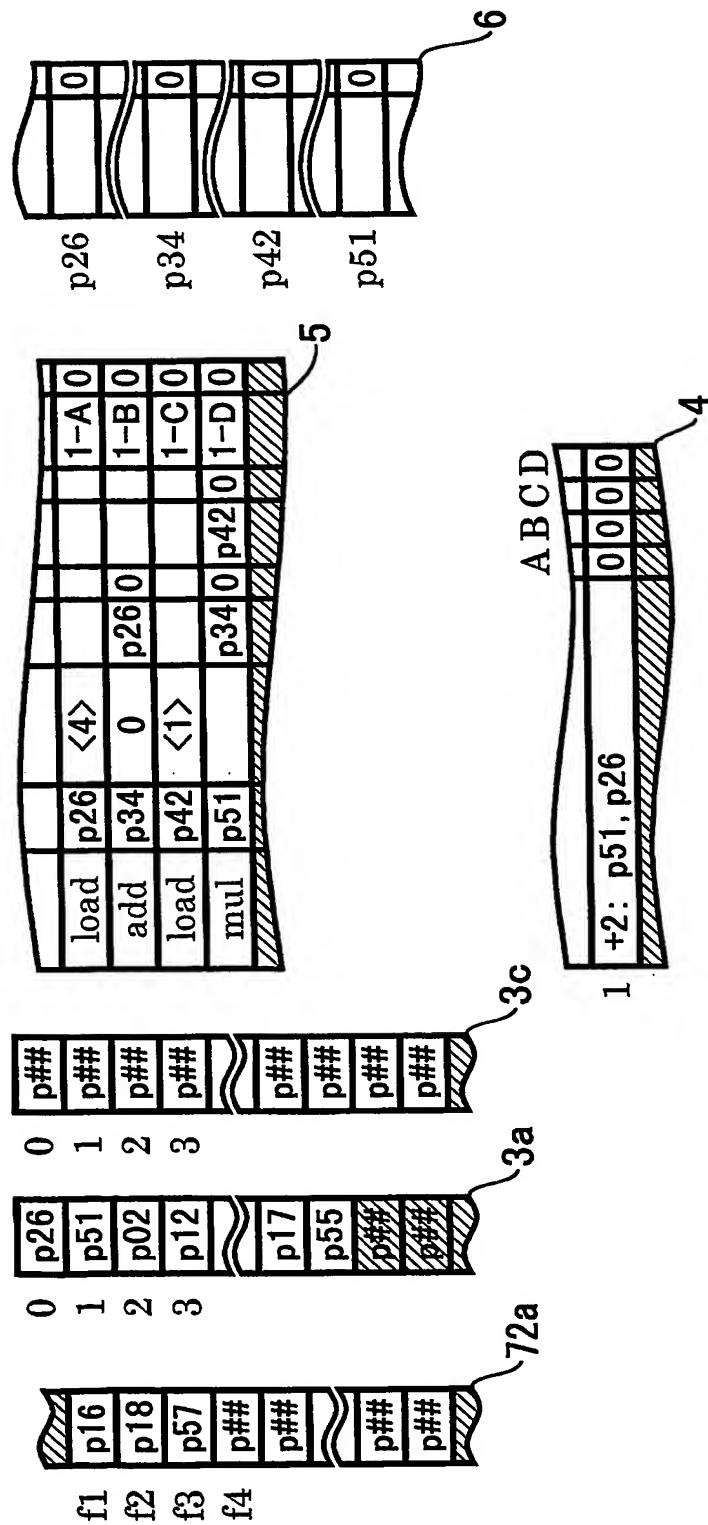
【図7】



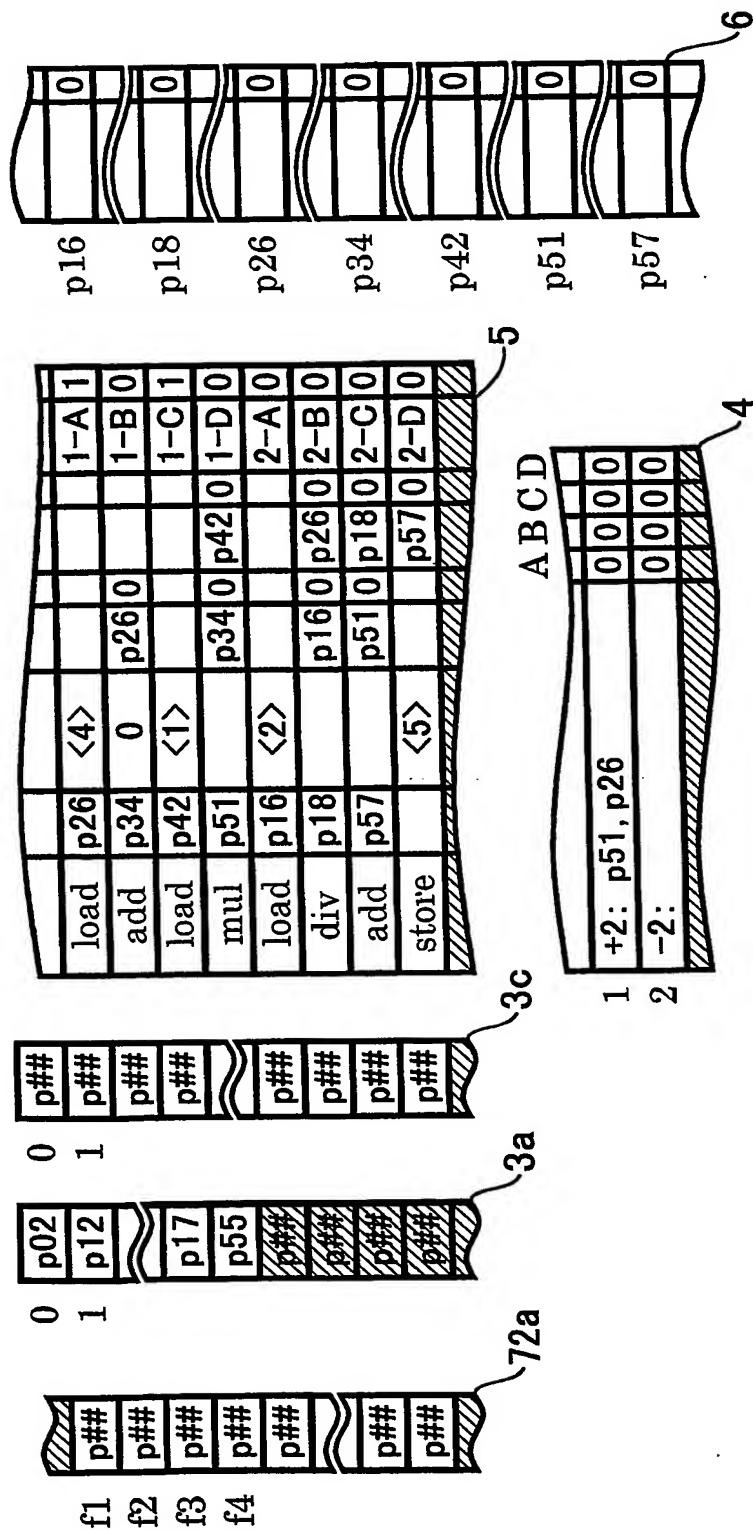
【図8】



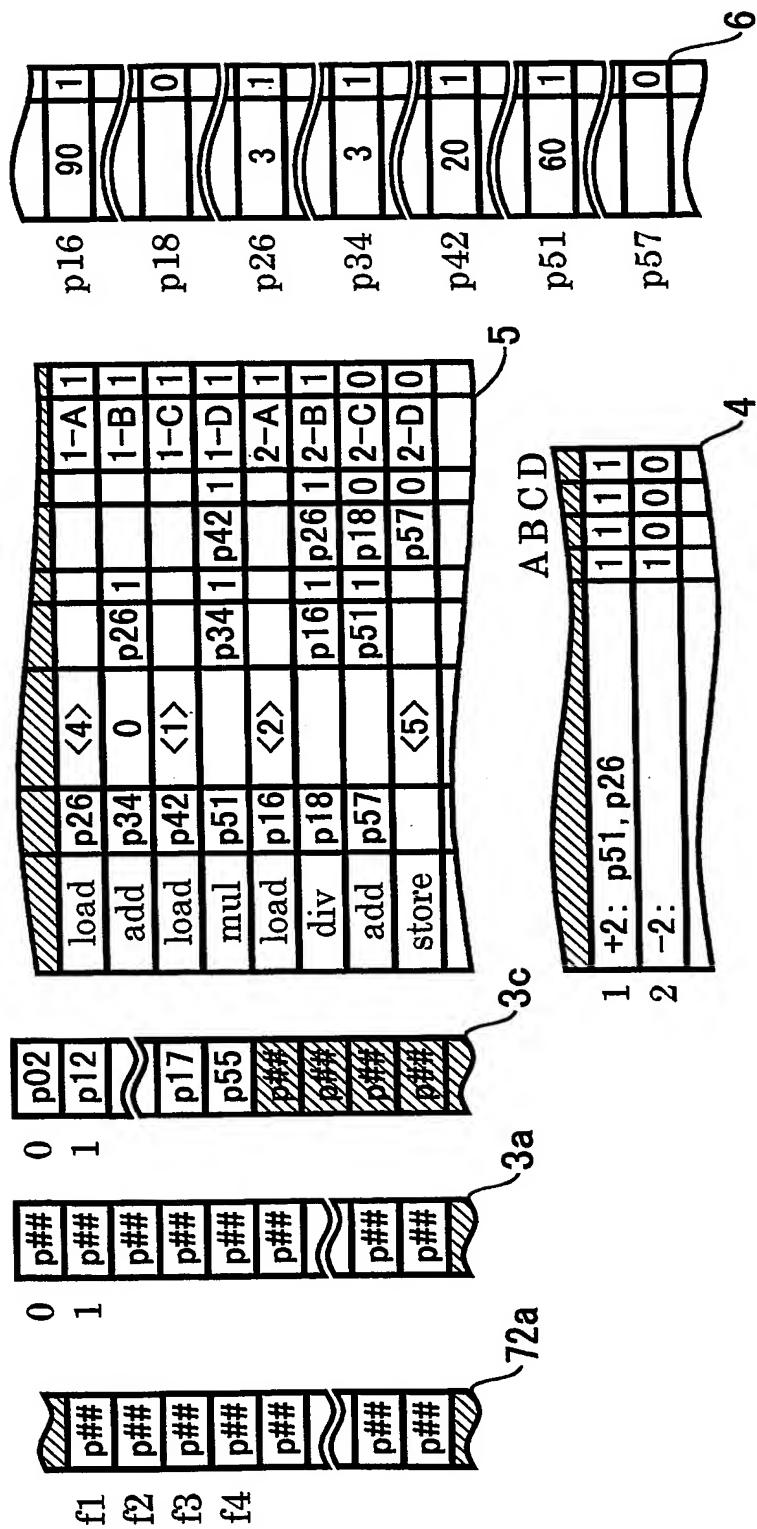
【図9】



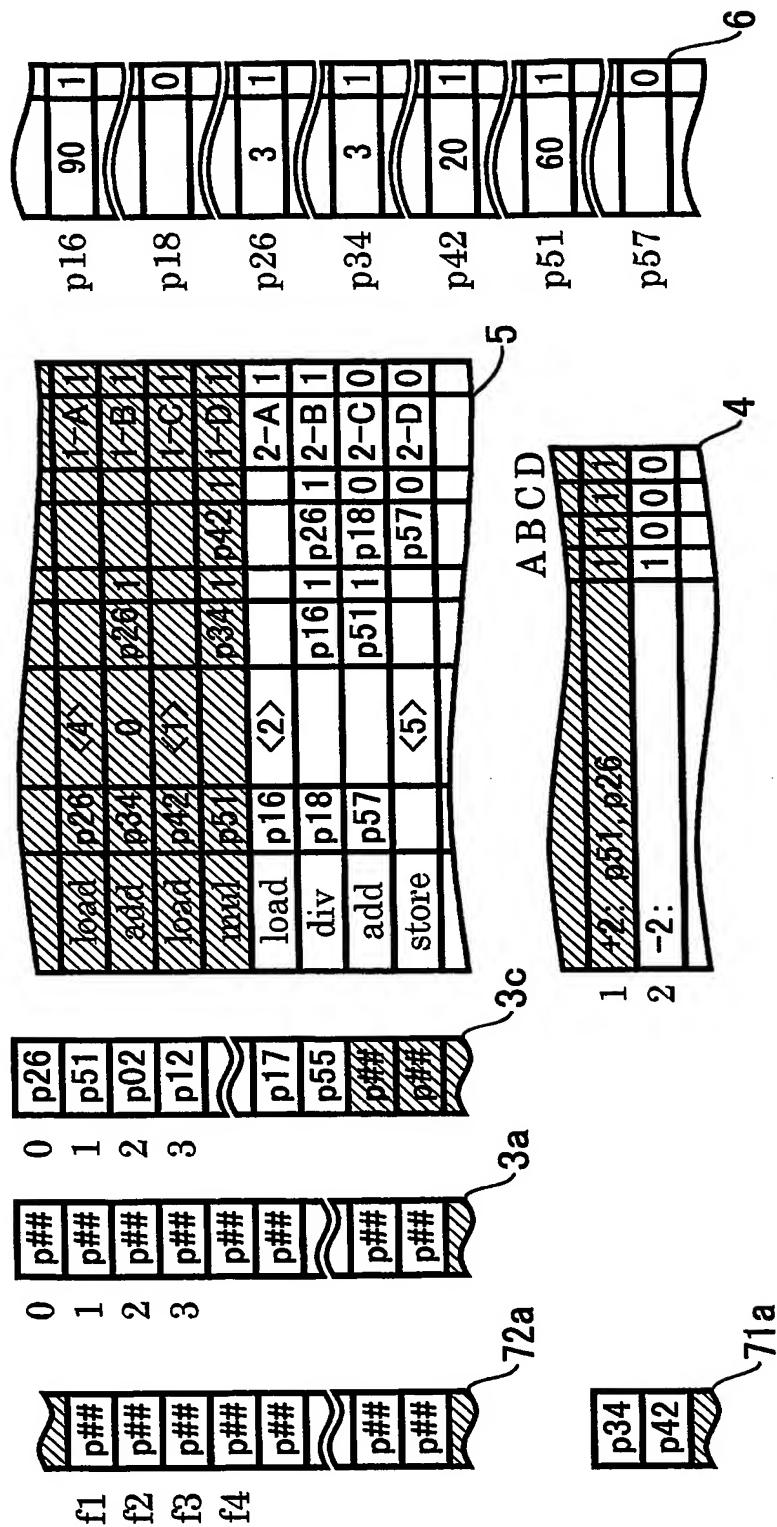
【図10】



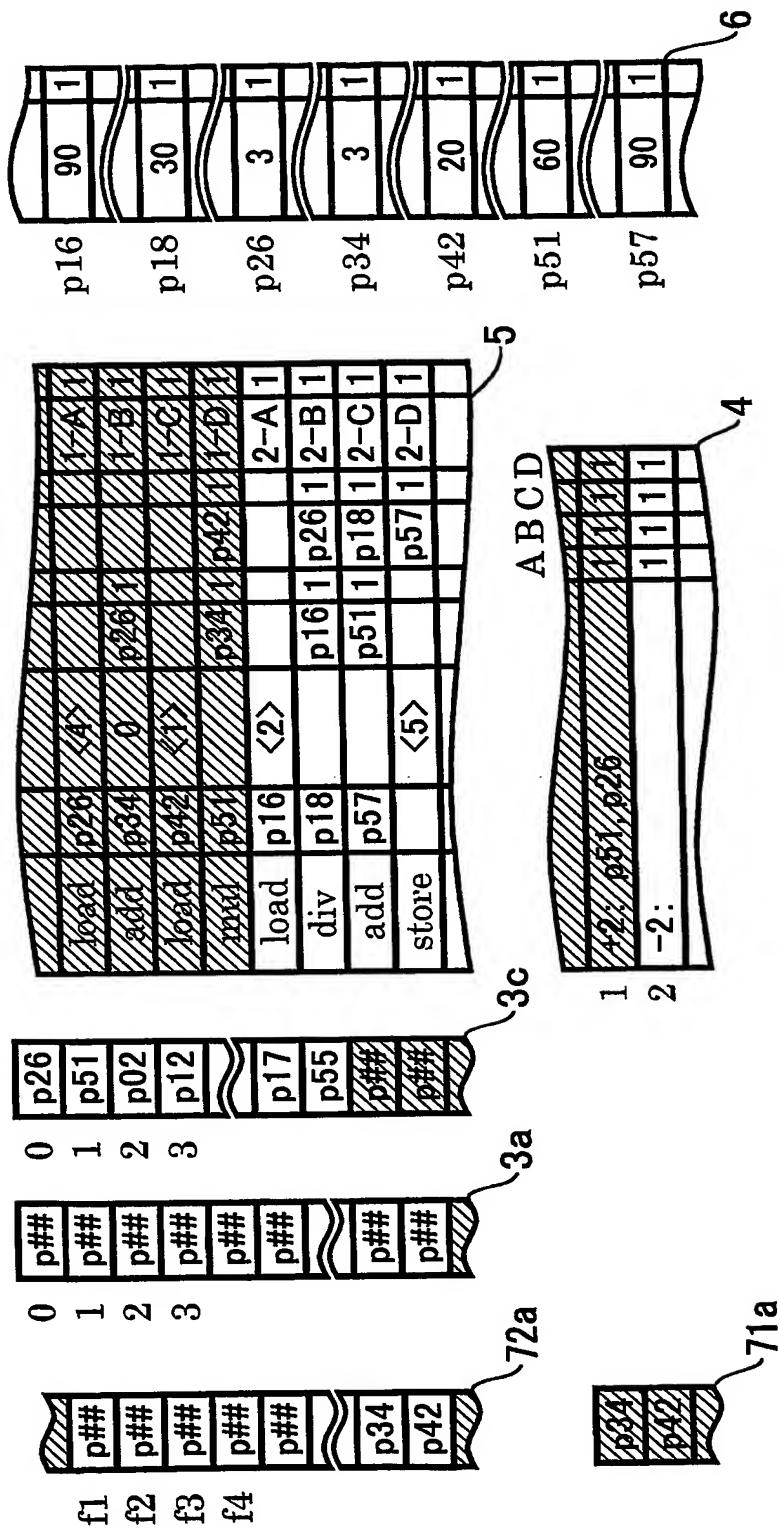
### 【図11】



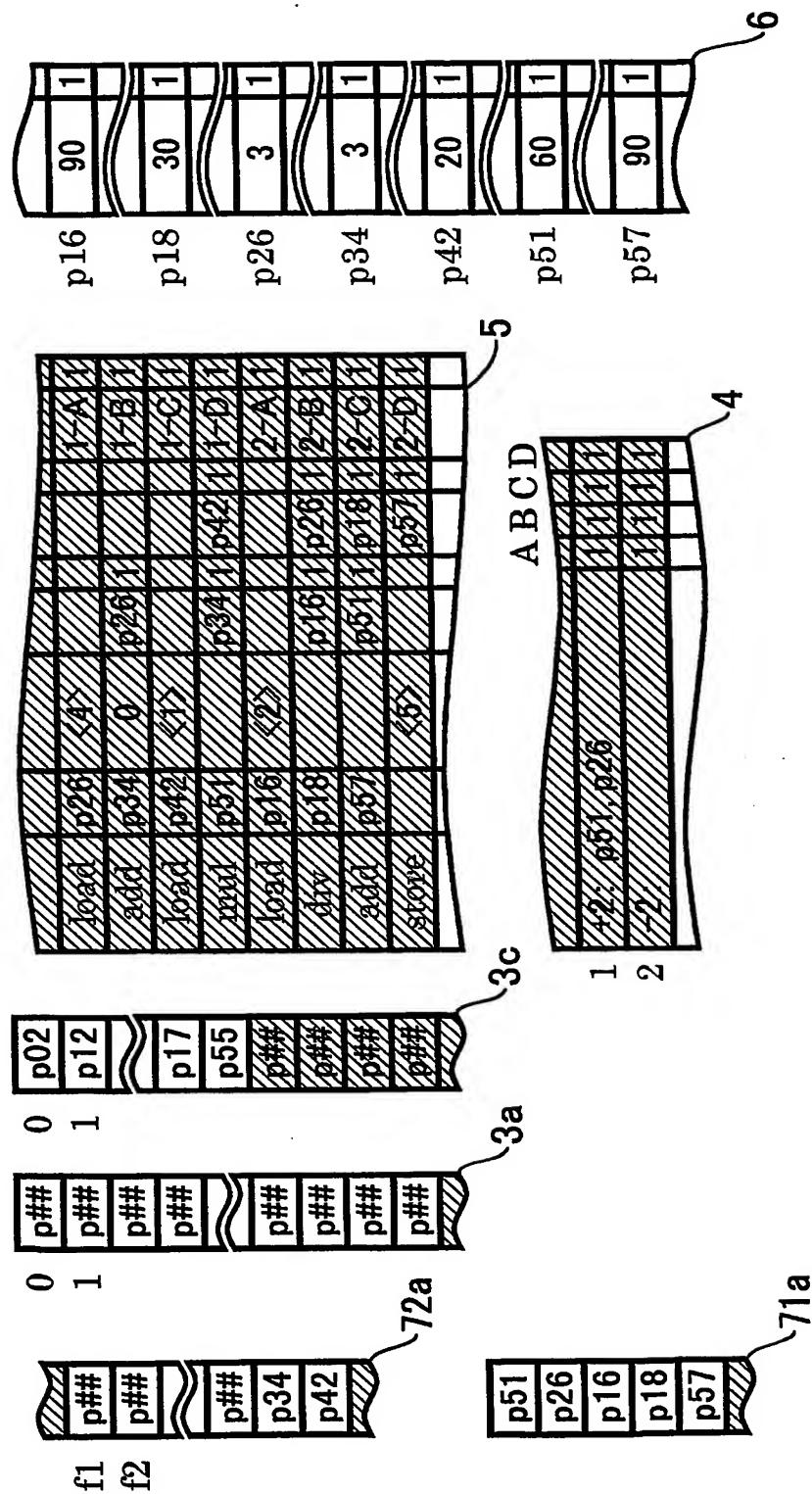
【図12】



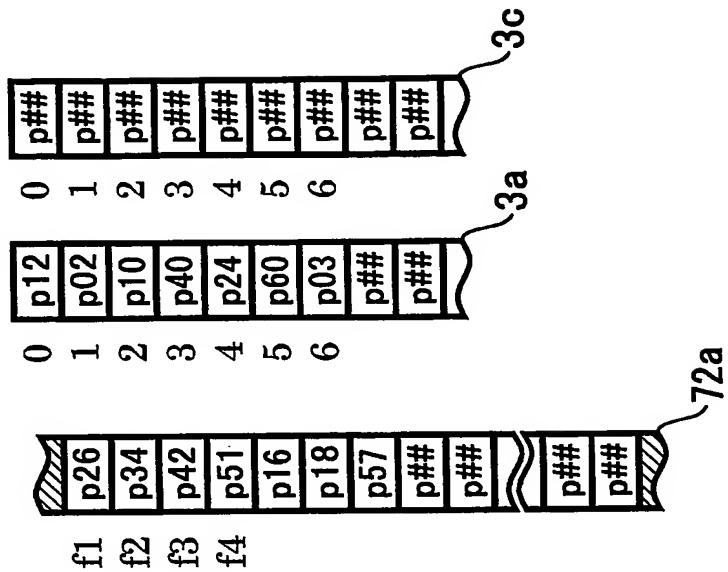
【図13】



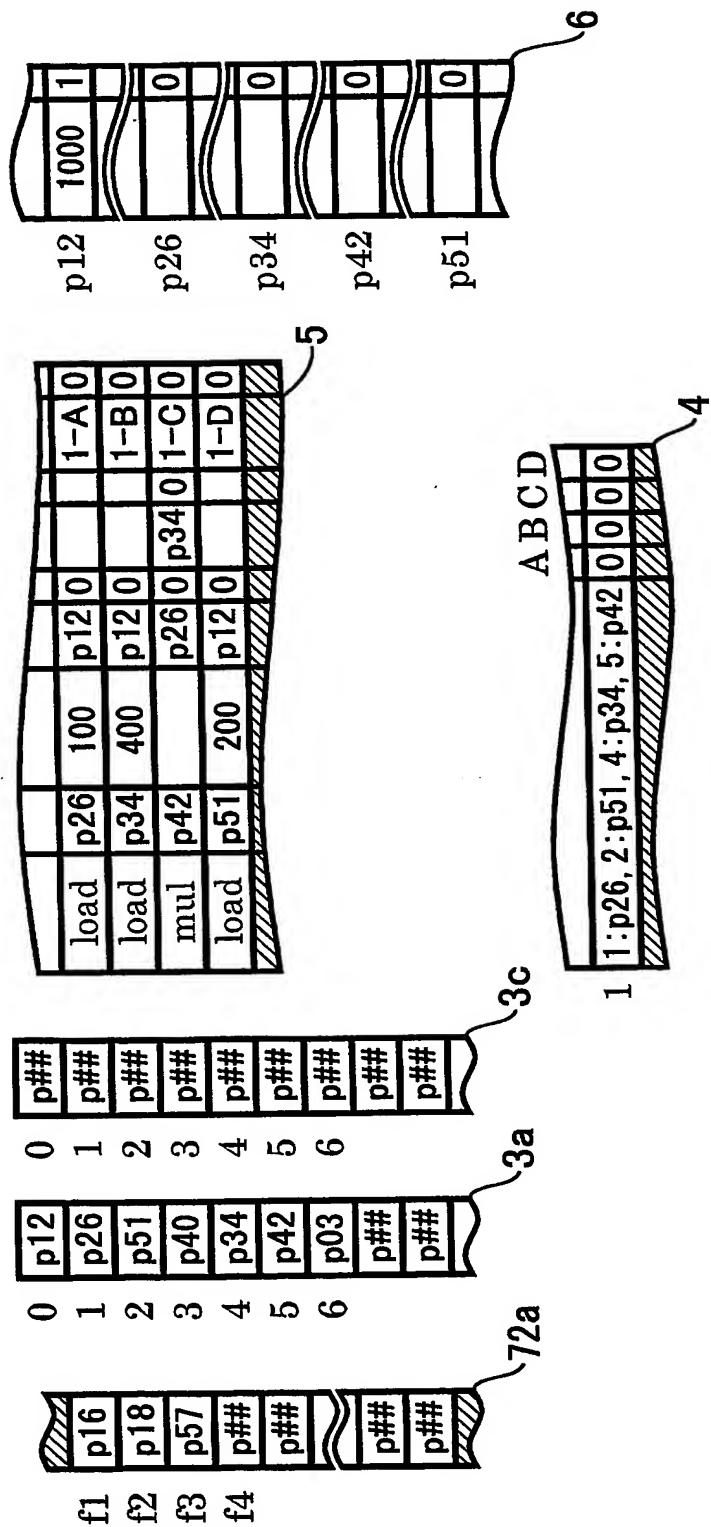
【図14】



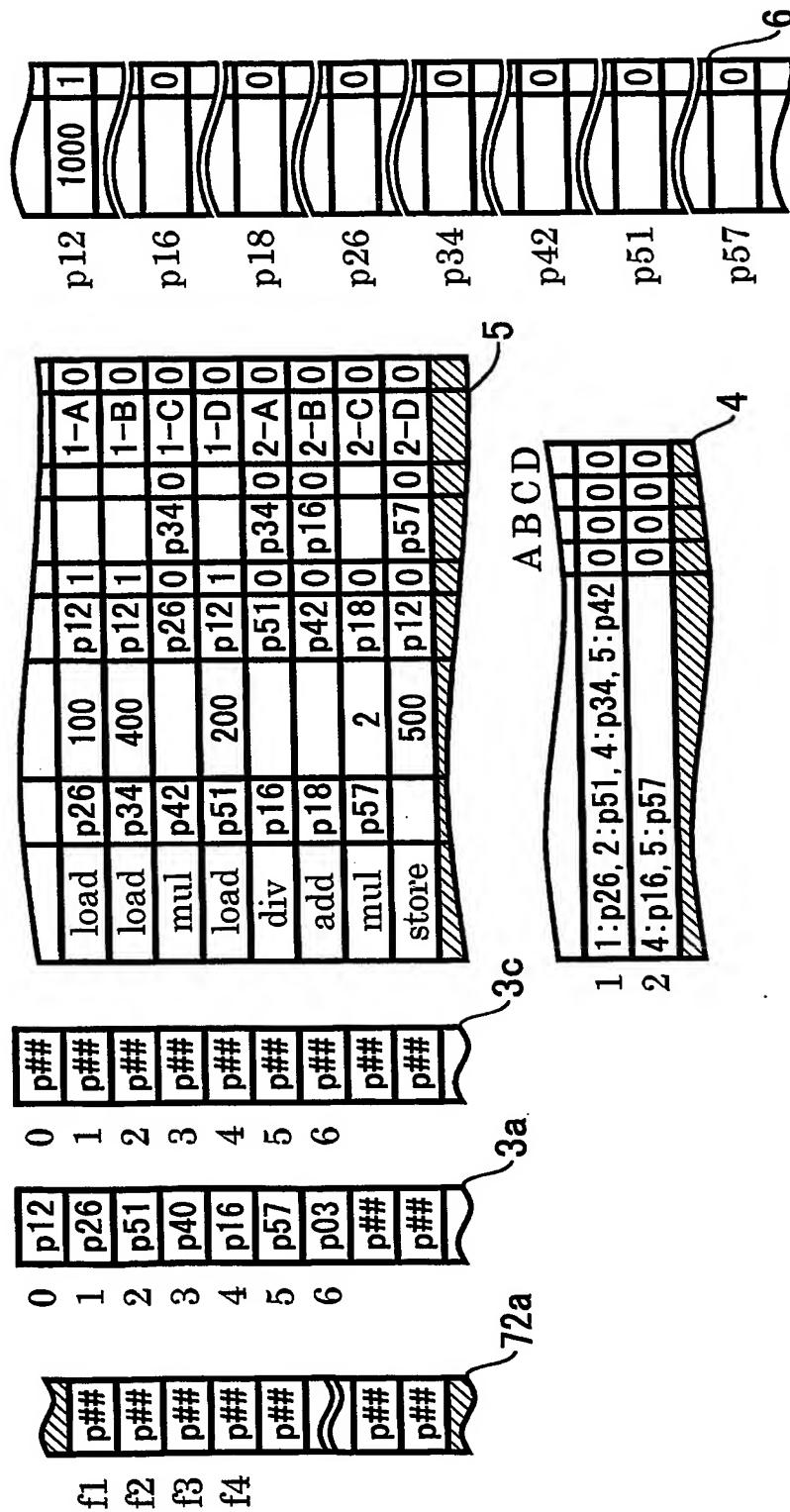
【図15】



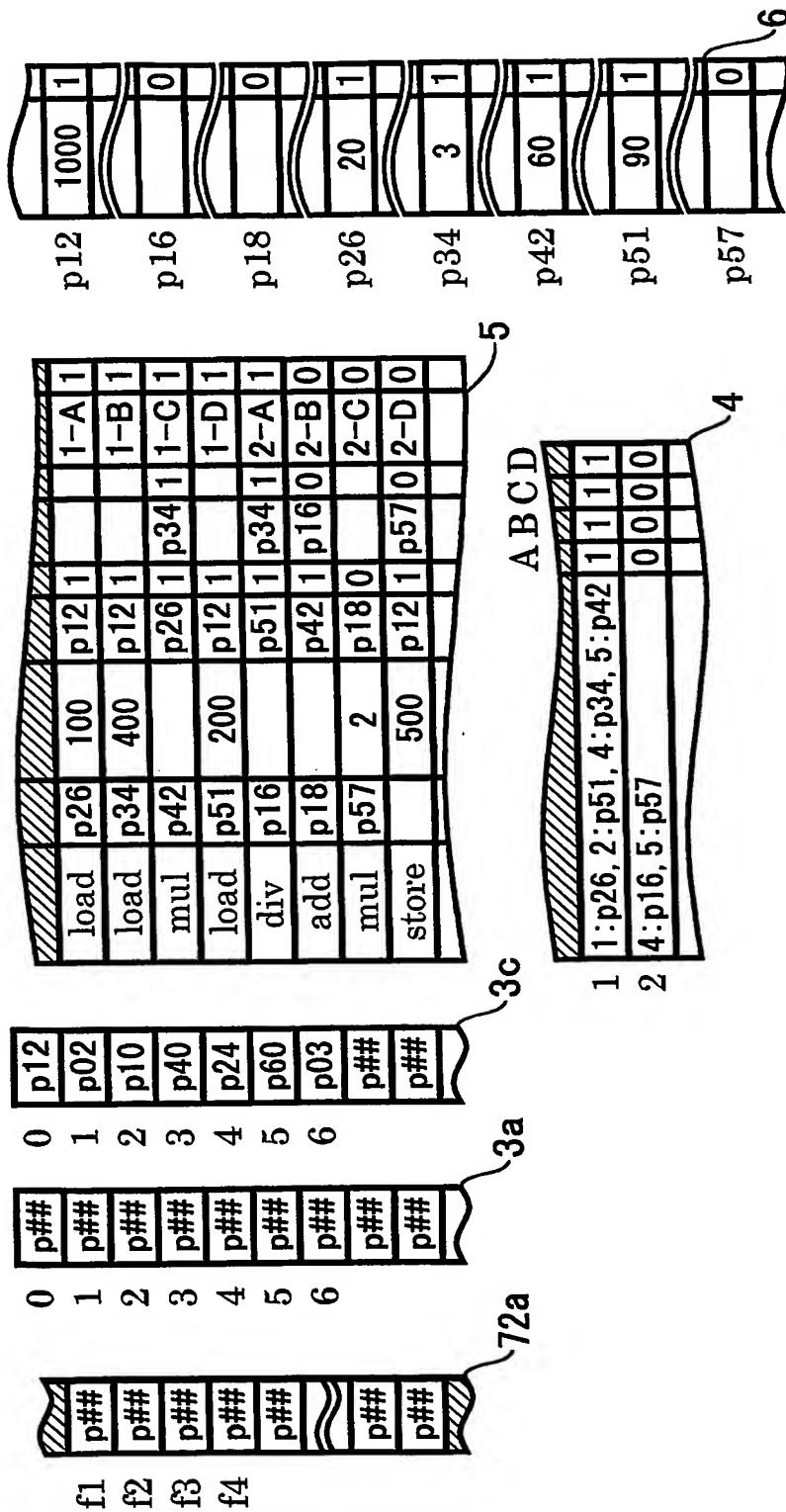
【図16】



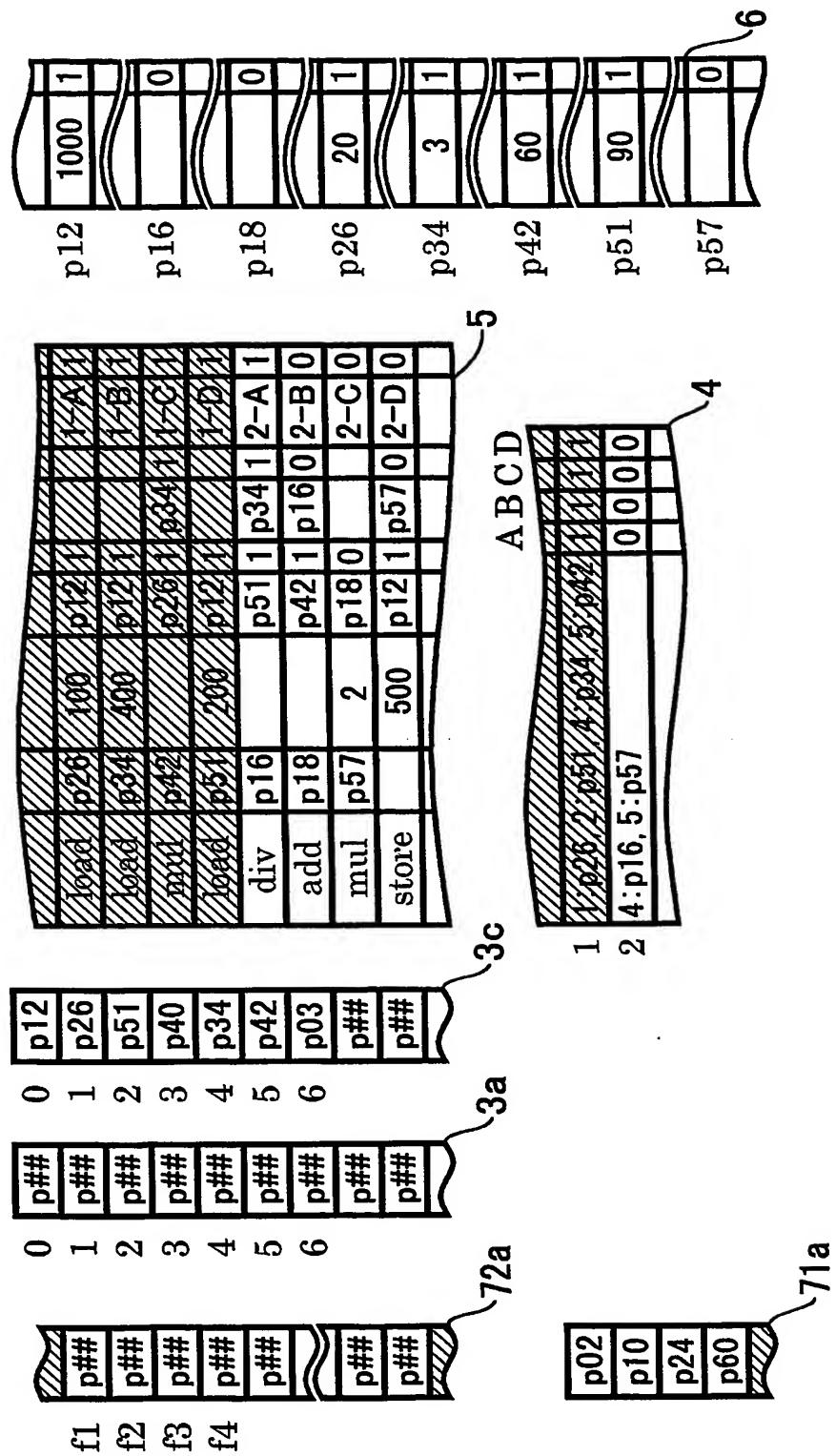
【図17】



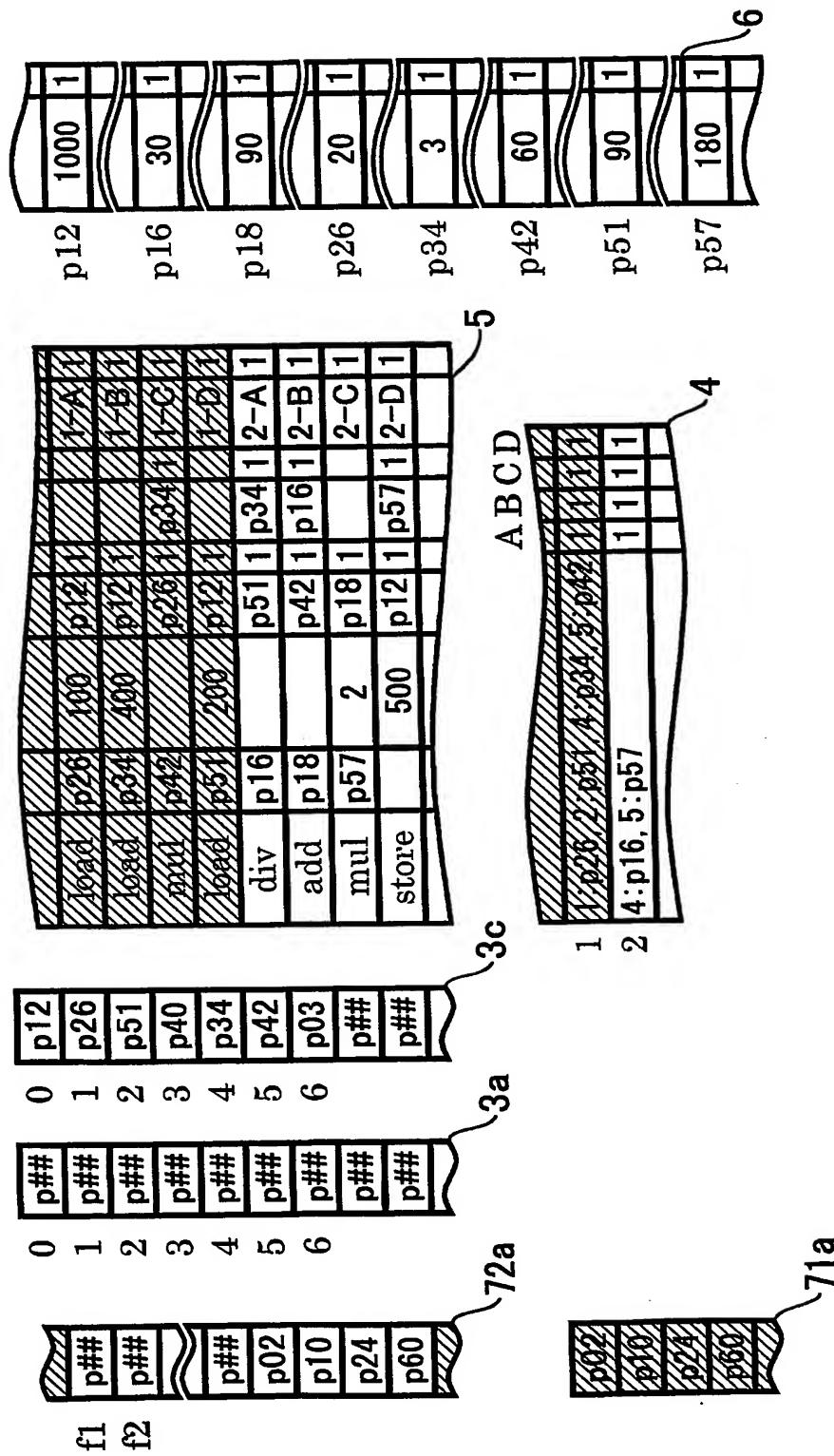
### 【図18】



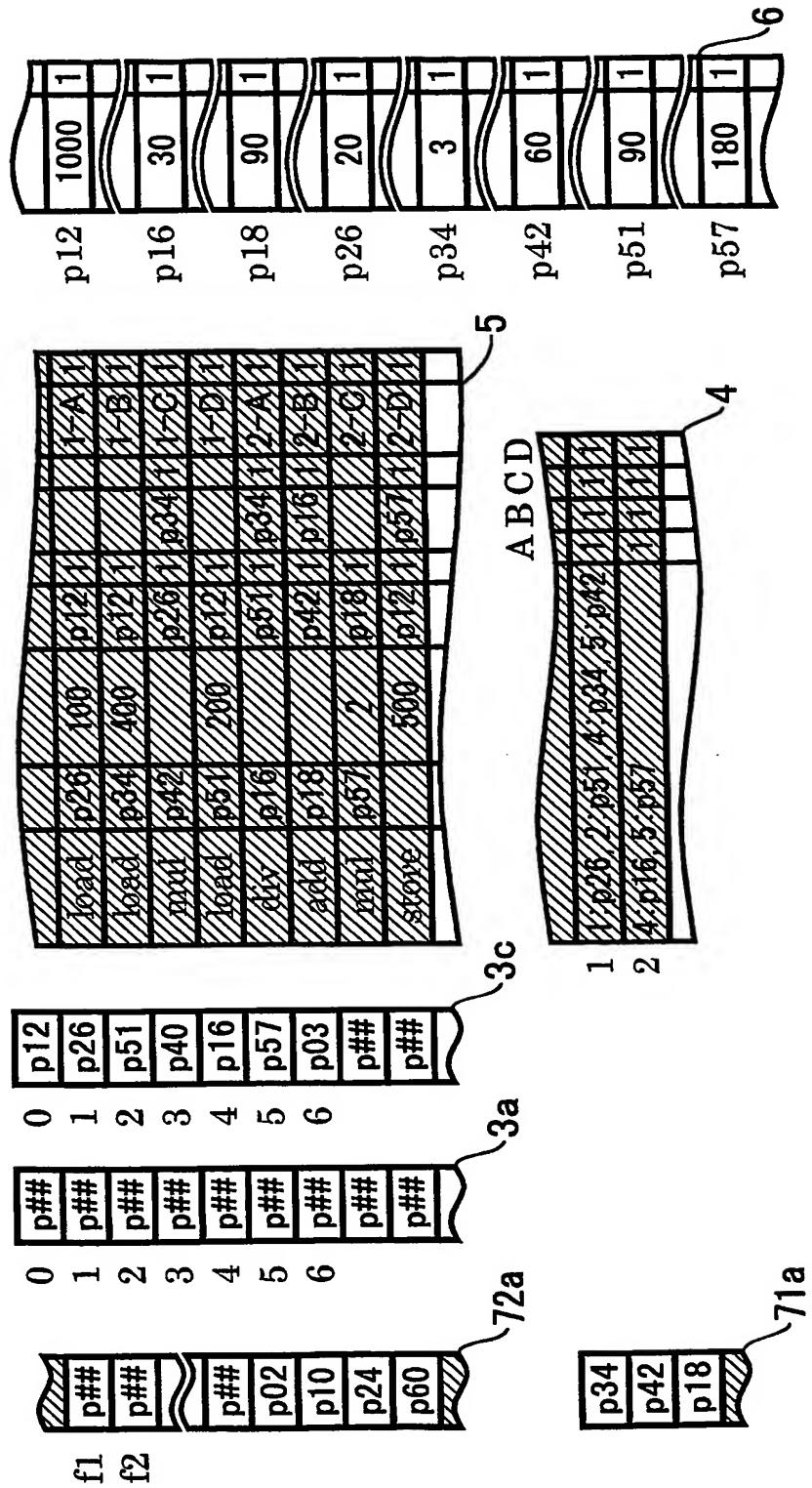
【図19】



【図20】



【図21】



【書類名】 要約書

【要約】

【目的】 高性能の計算機システムを実現する。

【構成】 各々のエントリにデータが書き込まれるようになっているデータ・ファイル6と、各々のエントリにデータ・ファイルのエントリのアドレスが書き込まれるようになっている前進マッピング・ファイル3a及び完了マッピング・ファイル3cと、各々のエントリに個々のオペレーションの内容が書き込まれるようになっているオペレーション・キュー5と、クロック・サイクル毎のマッピング・ファイルの状態変更の内容が書き込めるようになっている状態変更キューとを具備し、1クロック・サイクルで、複数のオペレーションの内容のオペレーション・キューへの書き込み、前進マッピング・ファイルの内容変更を行い、その変更の内容を状態変更キューに書き込む一方、オペレーション・キューに保持されるオペレーションがout-of-orderで実行される構成となっている。

【選択図】 図1

## 認定・付加情報

特許出願の番号	特願2002-254010
受付番号	50201298385
書類名	特許願
担当官	佐々木 吉正 2424
作成日	平成14年 9月 3日

## &lt;認定情報・付加情報&gt;

【提出日】	平成14年 8月30日
-------	-------------

次頁無

特願2002-254010

出願人履歴情報

識別番号 [598003070]

1. 変更年月日 1997年11月20日

[変更理由] 新規登録

住所 愛媛県松山市道後喜多町4番38号  
氏名 関一